NASA

NASA LAUNCH VEHICLE OPTIMIZATION

N66-14951
(ACCESSION NUMBER) (THRU) 1
627
(PAGES) (CODE)
CR-69212
(NASA CR OR TMX OR AD NUMBER) 31 (CATEGORY)
FACILITY FORM 602

LAUNCH VEHICLE OPTIMIZATION

GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) 9.27

Microfiche (MF) 3.00

ff 653 July 65

PHASE II FINAL REPORT

Volume II

Techniques Development

Lear Siegler, Inc.

LEAR SIEGLER PUBLICATION NO. GRR-65-1089-II
to
Office of Launch Vehicle and Propulsion Program
NASA Headquarters
Washington, D.C.

LAUNCH VEHICLE OPTIMIZATION

PHASE II FINAL REPORT

September 1965

Prepared by
Instrument Division
Under
Contract NASw-938

LEAR SIEGLER, INC.
INSTRUMENT DIVISION
4141 EASTERN AVENUE, S.E.
GRAND RAPIDS, MICHIGAN 49508

ABSTRACT

VOLUME II

TECHNIQUES DEVELOPMENT - LEAR SIEGLER

This second volume of a three-volume final report discusses results of a number of integral programs undertaken at LSI in developing the PRESTO concept. It contains six separate appendices -- each covering one of the following technique developments at LSI:

- Performance
- Reliability
- Economics
- Launch Vehicle System
- Regression
- Optimization

With the exception of Launch Vehicle System, each appendix presents results of one particular technique development as related to the overall PRESTO concept synthesis.

The Performance, Reliability, and Economics appendices are concerned with the system simulation techniques necessary to generate mathematical models describing these particular system characteristics. The models are required to have some independent system variables in common for overall concept implementation.

The Launch Vehicle System appendix is devoted to the definition of a Study Launch Vehicle System (SLVS) to which a modification of the PRESTO concept was applied for demonstration purposes.

The Regression appendix presents an approach to multiple regression curve fitting of data on a set of system parameters in order to generate an expression describing the behavior of one of the parameters in terms of the remaining (independent) parameters. This technique is intended for use in conjunction with any one or all of the Performance, Reliability, and Economics simulators in developing mathematical models of the various system characteristics.

The Optimization appendix presents a method which may be employed to exercise the earlier described models, leading to desired optimal system characteristics.

FOREWORD

VOLUME II

TECHNIQUES DEVELOPMENT - LEAR SIEGLER

A final report in three Volumes is herewith submitted by Lear Siegler, Inc., Instrument Division to the National Aeronautics and Space Administration Headquarters in fulfillment of Contract NASw-938. The study, entitled Launch Vehicle Optimization Study -- Phase II, was pursued under the technical direction of the Launch Vehicle and Propulsion Program Office, Code SV, NASA Headquarters, by the following participant organizations:

- LSI Instrument Division
- LSI Defense Systems Operations
- The University of Michigan

This summary report of the Phase II Launch Vehicle Optimization Study is contained in three separate volumes.

• Volume I Concept Development And Application	Volume I contains a general review of the program, an exposition of the PRESTO concept and techniques, a presentation of its application to a Study Launch Vehicle System, and a discussion of special problems and significant achievements.
• Volume II Techniques Development - Lear Siegler, Inc.	Volume II contains a comprehensive review of the PRESTO simulation and optimization techniques, as formulated and applied at LSI, in addition to a description of the Study Launch Vehicle System to be analyzed.
• Volume III Techniques Development - University of Michigan	Volume III contains a report of related efforts compiled by the University of Michigan under the direction of Dr. Frank H. Westervelt. It includes documentation on the Westervelt Performance Simulator and the U of M Regression Routine.

APPENDIX A

PERFORMANCE

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1 INTRODUCTION

Of increasing interest to systems analysts is the use of digital computers for system simulation. Historically, the analog computer has been the primary tool in machine simulation of a system. As the speed and flexibility of digital machines has improved, the use of digital computers has become more appealing. The reason for this is that for many types of problems, digital simulation can provide more reliable and accurate results with less over-all engineering time and effort.

The trend towards use of digital computers for system simulation has resulted in the expenditure of considerable effort towards developing digital computer programs capable of translating analog computer simulations directly to digital computer form. Since the first attempt by Selfridge [1] in 1955, there have appeared a number of such programs; best known perhaps are DEPI[2], ASTRAL[3], DEPI 4[4], DYSAC[5], PARTNER[6], DAS[7], JANIS[8], MIDAS[9], and PACTOLUS[10].

A basic requirement for the above mentioned programs is that the set of system equations (or state model) has been developed by the systems analyst in a form suitable for analog computer simulations. This state model development comprises a considerable portion of the overall system simulation effort. The use of a digital computer to develop the state model -- automatically -- constitutes a major breakthrough in the system simulation area. This is the strong point of the MISSAP* program discussed in Section 2 of this Appendix.

Once a system state model has been constructed, calculation of the system variables for a given set of component parameter values and input variables becomes efficient and accurate. However, the construction of a true performance model as conceived during the PRESTO concept development requires a considerable extension of the state model construction discussed in Section 2.

This extension involves the following considerations. Thorough analysis of system performance generally involves many repeated calculations of the system variable values for various combinations of component parameter values within their allowable ranges. This is required in order to assure that the system variables will lie within certain desired limits for

*The MIchigan State System Analysis Program (MISSAP) is a digital computer program being developed at Michigan State University under the sponsorship of International Business Machines Corporation.

the allowable ranges on the system component parameters. An increasingly popular approach to this problem involves Monte Carlo techniques. These techniques will yield a frequency distribution for each of the system variables for a given set of frequency distributions on the system component parameters. However, this approach is very tedious and time consuming. Construction of a true performance model involves a further extension of this approach in that a functional relationship is desired which explicitly relates the frequency distributions of the system variables to the frequency distribution of the system component parameters. With such a model, the effect of adjusting any or all of the component frequency distributions could be computed directly.

Three approaches to this problem are discussed in Section 3 of this Appendix. Each of these, essentially, consists of exercising the system state model discussed in Section 2. The merits of these approaches have not been thoroughly established and much work remains to be done in this area. However, it is felt that these may provide significant break-throughs in the area of system performance simulation.

Section 4 contains a list of references which were used in the concept development.

2 STATE MODEL GENERATION

This section deals primarily with the development of the system equations (referred to as the system "state model"). The state model is developed by the digital computer through use of various aspects of linear graph theory.

Section 2.1 explains in detail the MISSAP state model formulation procedure.

Section 2.2 illustrates this formulation technique by means of a simple example using an electrical RLC network.

Section 2.3 describes some of the modifications that are necessary to include additional components into the formulation procedure. The present version of MISSAP is restricted primarily to linear electrical networks. However, the general state model formulation technique is extendable to include non-electrical and non-linear components.

Section 2.4 illustrates the simulation of a reasonably complex electrical system made up of components presently in the MISSAP library. These include a non-linear model for a transistor. This example demonstrates the capability of the state modeling technique and provides some insight into the possibilities offered by this method for launch vehicle simulation.

2.1 MISSAP FORMULATION PROCEDURE

The information on the type of components and their mode of interconnection is supplied to the computer via punched cards giving the component type, the network node numbers between which the component is connected, its orientation, and the particular coefficient values describing the components such as resistance, capacitance, amplitude, rise-time, initial conditions, etc. In addition, the user must supply the voltages and currents desired as outputs by specifying appropriate voltmeters and ammeters in the network. The remainder of the formulation and solution is conducted automatically by the digital computer.

Associated with each element is both a voltage and a current variable, but either one or the other of these may be eliminated by the computer at the outset since the terminal equations for the element could then supply the missing variable. For example, if a resistor's voltage is eliminated, the product of the resistance and the current could supply the missing variable.

The method of choosing which variables should be eliminated and the method of how the equations relating the remaining variables should be written to insure correct formulation has been developed by Reed and Seshu[11], Koening and Blackwell[12], Wirth[13], and Reid[14].

The technique of choosing which variables should be eliminated proceeds in the computer as follows.

The element-node incidence matrix is generated from the element-current summations at each of the nodes of the network as

$$\mathcal{a}\, I = 0 \qquad \text{Eq. 1}$$

where the rows of $\mathcal{a}$ correspond to nodes and the columns correspond to elements. Next, rows of $\mathcal{a}$ are added and/or subtracted and/or permuted and the columns of $\mathcal{a}$ are permuted. This forces the following form

$$\begin{bmatrix} U & S \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_B \\ I_C \end{bmatrix} = 0 \qquad \text{Eq. 2}$$

where U is the unit matrix and S is the matrix obtained by forcing this particular form.

The order of the variables in I_B, the branch currents, and I_C, the chord currents will, in general, be different from their order in I since the columns of $\mathcal{a}$ will have been permuted.

The form required for Equation 2 will not give a unique result. However, not all equations of this form will yield the desired matrix properties in the following development. The additional requirement which must be met in forming Equation 2 is that of determining which columns of $\mathcal{a}$ to permute in forming the unit matrix; certain columns, i.e. elements, must be preferred over others. This order of preference is as follows going from left to right in $\mathcal{a}$;

1. Voltage sources
2. Ammeters
3. Capacitors
4. Transmission Lines

5. Resistors
6. Inductances
7. Voltmeters
8. Current sources

In many cases, there are advantages to be gained as far as subsequent numerical accuracy is concerned if a finer gradation of preferences is made within the above groups. For example, the smaller resistances are given preference over the larger valued resistances.

Equation 2 will always contain at least one row of zeroes, since at least one node current summation is not independent of the rest. In general, there may be p parts to the circuit, that is, p separate sections which have no connection whatever, or are connected by mutual inductances, transmission lines, or dependent sources. In the general case there would be p rows of zeroes.

The rows of zeroes of Equation 2 may be discarded leaving

$$\begin{bmatrix} U & S \end{bmatrix} \begin{bmatrix} I_B \\ \\ I_C \end{bmatrix} = 0 \qquad \text{Eq. 3}$$

from which I_B is obtained as

$$I_B = -SI_C \qquad \text{Eq. 4}$$

Thus the currents of the elements having currents I_B, i.e. the branch currents, may be expressed in terms of the chord currents.

In a similar manner, the chord voltages can be expressed in terms of the branch voltages. It can be shown [12] that the columns of S provide the required circuit information and that

$$V_c = S^T V_B \qquad \text{Eq. 5}$$

where S^T is the transpose of S given in Equation 3.

Thus, if the branch currents and the chord voltages are eliminated by Equations 4 and 5, the remaining variables are the chord currents and the branch voltages.

When the equations are checked at this point, all the voltage sources and ammeters must be in the set of branches, and all the current sources and voltmeters must be in the chord set or the solution does not exist for this network.

Next, the circuit equations are written from the specifications using the variables determined above. These equations are

$$\frac{d}{dt} M X + A X + B E(t) = 0 \qquad \text{Eq. 6}$$

where

$$X = \begin{bmatrix} V_B \\ I_C \end{bmatrix} \qquad \text{Eq. 7}$$

and E(t) is the set of driving functions specified. The coefficient matrices -- M, A, and B -- are implied by the specifications and generated by the program.

The derivatives of all the variables in X are, in general, not required, that is, some of them are simply algebraic variables. Likewise, all the equations in Equation 6 are not differential equations -- some are algebraic. If Equation 6 is rewritten in partitioned form, this separation of differential and algebraic equations and variables can be displayed as

$$\frac{d}{dt} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} X_d \\ X_a \end{bmatrix} + \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_d \\ X_a \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} E(t) = 0$$

Eq. 8

where the X_d are the variables whose derivatives are required and the X_a are the algebraic variables.

If Equation 8 has been properly partitioned, then the sub-matrix M_{11} will have maximum rank. Entries within M_{21} indicates that some of the capacitors and inductors will not require differential equations because of their interconnection in the network. For example, if three capacitors form a circuit in a network, the time derivative and magnitude of two of the capacitor voltages are sufficient to determine the time derivative and magnitude of the third. Since the X_a also contains the variables associated with the elements that are specified drivers, entries in M_{12} and M_{22} indicate that the time derivatives of the corresponding drivers are required for the solution, e.g., circuits of only capacitors and voltage drivers.

The operations on Equation 8 proceed as follows. Entries in M_{12} and M_{22} are deleted and a new coefficient matrix for the derivatives of the specified drivers is formed. The first equation is then multiplied by M_{11}^{-1}. The resulting new first equation is then mulitiplied by M_{21} and subtracted from the second equation. The equation then is

$$\frac{d}{dt}\begin{bmatrix} U & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} X_d \\ X_a \end{bmatrix} + \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & \mathcal{A}_{22} \end{bmatrix}\begin{bmatrix} X_d \\ X_a \end{bmatrix} + \begin{bmatrix} \mathcal{B}_1 \\ \mathcal{B}_2 \end{bmatrix} E(t) + \begin{bmatrix} \mathcal{C}_1 \\ \mathcal{C}_2 \end{bmatrix} E'(t) = 0$$

Eq. 9

The solution to Equation 9 exists only if $\mathcal{A}_{22}^{-1}$ exists, so the second equation of (9) may be multiplied by $\mathcal{A}_{22}^{-1}$. This effects the solution for the X_a in terms of the X_d, and X_a may be replaced in the first equation. The form then is

$$\frac{d}{dt} X + \mathcal{A} X + \mathcal{B} E(t) + \mathcal{C} E'(t) = 0 \qquad \text{Eq. 10}$$

Equation 10 is in a suitable form for numerical or analytic solution. This equation is solved numerically the the present version of MISSAP.

2.2 ILLUSTRATIVE EXAMPLE

The following example parallels exactly the preceding explanation of the MISSAP formulation technique.

Consider the electrical network shown in Figure A-1.

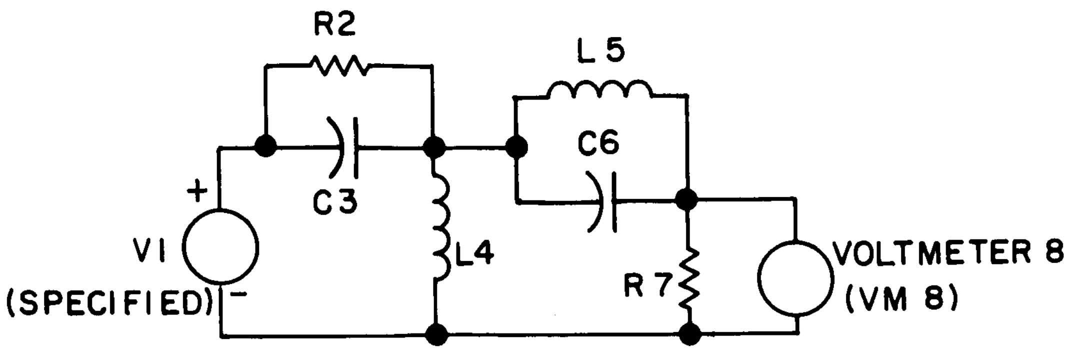


Figure A-1. Example RLC Network

A linear graph of the network is implied by the interconnection and orientation information that is supplied in the input specification by the user. This graph is given in Figure A-2.

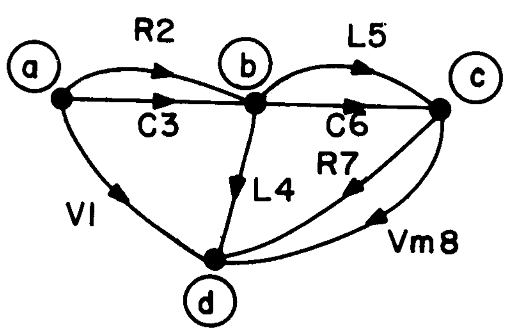


Figure A-2. Linear Graph for Example Problem

The element-node incidence matrix of Equation 1 is then automatically constructed by Kirchhoff's current law as

$$\begin{array}{cc}
 & \overbrace{\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array}}^{\text{Elements}} \\
\text{Nodes}\left\{\begin{array}{c} a \\ b \\ c \\ d \end{array}\right. & \begin{bmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
\begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \\ i_8 \end{bmatrix} = 0 \qquad \text{Eq. 11}$$

Note that the element subscript in Figure A-2 corresponds to the number given to the element in Equation 11.

To obtain the form that is given in Equation 2, the columns of the coefficient matrix of Equation 11 are first permuted so that the elements are in the proper order to satisfy the preference list, i.e.,

$$\begin{array}{cc}
 & \begin{array}{cccccccc} 1 & 3 & 6 & 2 & 7 & 4 & 5 & 8 \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}
\end{array}
\begin{bmatrix} i_1 \\ i_3 \\ i_6 \\ i_2 \\ i_7 \\ i_4 \\ i_5 \\ i_8 \end{bmatrix} = 0 \qquad \text{Eq. 12}$$

Adding rows a, b, c and d and replacing row d by the sum provides a bottom row of zeros, i.e.,

$$d' = a + b + c + d = 0$$

Performing the following operations

$$b' = b + c$$

$$a' = -a - b'$$

yields the desired form of Equation 2

$$\begin{array}{c} \begin{array}{ccc|ccccc} 1 & 3 & 6 & 2 & 7 & 4 & 5 & 8 \end{array} \\ \left[\begin{array}{ccc|ccccc} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & -1 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 & 1 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \left[\begin{array}{c} i_1 \\ i_3 \\ i_6 \\ \hline i_2 \\ i_7 \\ i_4 \\ i_5 \\ i_8 \end{array}\right] = 0 \qquad \text{Eq. 13}$$

Therefore

$$\begin{bmatrix} i_1 \\ i_3 \\ i_6 \end{bmatrix} = - \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & -1 & -1 & 0 & -1 \\ 0 & -1 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} i_2 \\ i_7 \\ i_4 \\ i_5 \\ i_8 \end{bmatrix} \qquad \text{Eq. 14}$$

which is the form of Equation 4 where

$$I_B = \begin{bmatrix} i_1 \\ i_3 \\ i_6 \end{bmatrix} \qquad \text{Eq. 15}$$

and

$$I_C = \begin{bmatrix} i_2 \\ i_7 \\ i_4 \\ i_5 \\ i_8 \end{bmatrix} \qquad \text{Eq. 16}$$

From Equation 5

$$V_C = \begin{bmatrix} v_2 \\ v_7 \\ v_4 \\ v_5 \\ v_8 \end{bmatrix} = S^T V_B = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & -1 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_3 \\ v_6 \end{bmatrix} \qquad \text{Eq. 17}$$

which can be verified by Figures A-1 and A-2.

The form of the equations that are stored in MISSAP to describe the components that are used in the example is as follows:

v_1 = a function of time determined by the input specification. (Note: There are presently six time functions available for use as voltage drivers. A seventh can be used to supply additional functions by means of data cards.)

$$v_2 = R_2 i_2$$

$$\frac{dv_3}{dt} = \frac{1}{C_3} i_3$$ (an initial value of v_3 is part of the input specification)

$$\frac{di_4}{dt} = \frac{1}{L_4} v_4$$

$$\frac{di_5}{dt} = \frac{1}{L_5} v_5 \qquad \text{Eq. 18}$$

$$\frac{dv_6}{dt} = \frac{1}{C_6} i_6$$

$$v_7 = R_7 i_7$$

$$i_8 = 0$$

Eliminating the branch currents and the chord voltages that are given by Equations 14 and 17, and rewriting the results in matrix form provides

$$
\frac{d}{dt}\begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0 \end{bmatrix}\begin{bmatrix} v_1\\ v_3\\ v_6\\ i_2\\ i_7\\ i_4\\ i_5\\ i_8 \end{bmatrix} =
$$

Eq. 19

$$
\begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&\frac{-1}{C_3}&\frac{1}{C_3}&\frac{1}{C_3}&0&\frac{1}{C_3}\\ 0&0&0&0&\frac{1}{C_6}&0&\frac{-1}{C_6}&\frac{1}{C_6}\\ 0&1&0&-R_2&0&0&0&0\\ 1&-1&-1&0&-R_7&0&0&0\\ \frac{1}{L_4}&\frac{-1}{L_4}&0&0&0&0&0&0\\ 0&0&\frac{1}{L_5}&0&0&0&0&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}\begin{bmatrix} v_1\\ v_3\\ v_6\\ i_2\\ i_7\\ i_4\\ i_5\\ i_8 \end{bmatrix} + \begin{bmatrix} -1\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0\\ 0 \end{bmatrix} e(t)
$$

where $e(t)$ is the specified function for v_1. This is the form of Equations 6 and 7.

Separating the algebraic equations from the differential equation provides the form given in Equation 8.

$$
\frac{d}{dt}\left[\begin{array}{cccc|cccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\left[\begin{array}{c}
v_3 \\ v_6 \\ i_4 \\ i_5 \\ \hline v_1 \\ i_2 \\ i_7 \\ i_8
\end{array}\right] =
$$

Eq. 20

$$
\left[\begin{array}{cccc|cccc}
0 & 0 & \frac{1}{C_3} & 0 & 0 & \frac{-1}{C_3} & \frac{1}{C_3} & \frac{1}{C_3} \\
0 & 0 & 0 & \frac{-1}{C_6} & 0 & 0 & \frac{1}{C_6} & \frac{1}{C_6} \\
\frac{-1}{L_4} & 0 & 0 & 0 & \frac{1}{L_4} & 0 & 0 & 0 \\
0 & \frac{1}{L_5} & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & -R_2 & 0 & 0 \\
-1 & -1 & 0 & 0 & 1 & 0 & -R_7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right]
\left[\begin{array}{c}
v_3 \\ v_6 \\ i_4 \\ i_5 \\ \hline v_1 \\ i_2 \\ i_7 \\ i_8
\end{array}\right]
+
\left[\begin{array}{c}
0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0
\end{array}\right] e(t)
$$

Note that for this particular example

$$M_{12} = M_{21} = M_{22} = 0$$

Hence, the derivative of e(t) is not necessary and the number of differential equations in Equation 20 cannot be reduced. The bottom half of Equation 20 provides

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -R_2 & 0 & 0 \\ 1 & 0 & -R_7 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ i_2 \\ i_7 \\ i_8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ +1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_3 \\ v_6 \\ i_4 \\ i_5 \end{bmatrix} + \begin{bmatrix} +1 \\ 0 \\ 0 \\ 0 \end{bmatrix} e(t)$$

Eq. 21

or by premultiplying this equation by the inverse of the coefficient matrix on the left hand side of Equation 21

$$\begin{bmatrix} v_1 \\ i_2 \\ i_7 \\ i_8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{R_2} & 0 & 0 & 0 \\ \frac{-1}{R_7} & \frac{-1}{R_7} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_3 \\ v_6 \\ i_4 \\ i_5 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \frac{1}{R_7} \\ 0 \end{bmatrix} e(t) \qquad \text{Eq. 22}$$

Substituting Equation 22 into the top half of Equation 20 provides the desired state model given in Equation 10

$$
\frac{d}{dt}\begin{bmatrix} v_3 \\ v_6 \\ i_4 \\ i_5 \end{bmatrix} = \begin{bmatrix} -\left(\frac{1}{C_3R_2} + \frac{1}{C_3R_7}\right) & -\frac{1}{C_3R_7} & \frac{1}{C_3} & 0 \\ -\frac{1}{C_6R_7} & -\frac{1}{C_6R_7} & 0 & -\frac{1}{C_6} \\ -\frac{1}{L_4} & 0 & 0 & 0 \\ 0 & \frac{1}{L_5} & 0 & 0 \end{bmatrix} \begin{bmatrix} v_3 \\ v_6 \\ i_4 \\ i_5 \end{bmatrix} + \begin{bmatrix} \frac{1}{C_3R_7} \\ \frac{1}{C_6R_7} \\ \frac{1}{L_4} \\ 0 \end{bmatrix} e(t)
$$

Eq. 23

The numerical solution of these differential equations, along with Equations 14, 17, and 22 provide sufficient information to determine the time response of any variable of the system given in Figure A-1.

2.3 POSSIBLE EXTENSIONS

The extension of the present version of MISSAP to mechanical, electro-mechanical, hydraulic and other systems that have direct linear electrical analogs is trivial. That is, any system that can be simulated on a linear analog computer can also be simulated with the present version of MISSAP by simply including the additional equations that describe the non-electrical components. This type of extension will not be discussed further.

The use of topological linear graphs and complementary "through" and "across" variables for the description of multi-terminal electrical, mechanical, and hydraulic components has been described by Koenig and Blackwell.[12] This technique provides a basis for formulating the state model of a general linear system in the same fashion that is presently used in MISSAP for two terminal electrical components. The properties of the branch and chord sets that are used with this general technique are identical to those that are presently exploited in MISSAP and result in nearly the same matrix algebra that is described in the previous section. While there are some subtleties involved in the formulation of a state model with multiterminal components, these areas are well defined and can certainly be included in a computer program.

The formulation of a state model for systems composed of non-linear components (especially those containing trigonometric functions) is necessary to completely simulate a launch vehicle by this technique. Fortunately, the state model formulation for these non-linear systems exactly parallels the formulation process that is used in MISSAP through Equation 8 of Section 2.1, with Equations 6 and 8 replaced by first order non-linear differential equations. At this point, however, there is no guarantee that the resulting algebraic equations, which are now non-linear in nature, can be explicitly solved for the desired variables to obtain the final state model form. In cases where the "inversion" of these non-linear algebraic equations is impossible, the complete system of equations

$$\frac{d}{dt} x_d = f(x_d, x_a, E(t)) \qquad \text{Eq. 24}$$

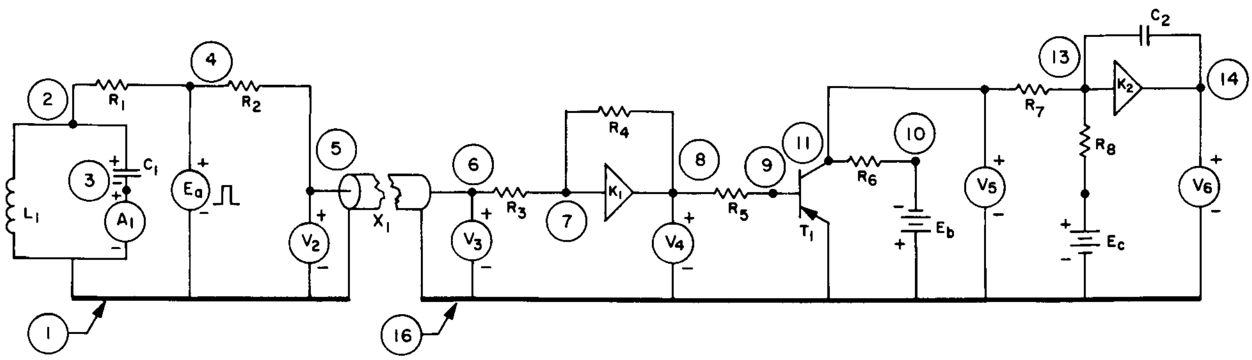
$$0 = g(x_d, x_a, E(t)) \qquad \text{Eq. 25}$$

(where g is the system of non-linear algebraic equations that cannot be explicitly solved for x_a in terms of $E(t)$ and x_d) is solved numerically in the following fashion. Initial conditions $x_d\,(t=0)$ are prescribed for the dynamic elements of the system. Since the values of $E(t=0)$ and $x_d(t=0)$ are known, Equation 25 can be solved by numerical iteration for $x_a(t=0)$. This provides the necessary information for the evaluation of $\frac{d\,x_d}{dt}$ in Equation 24 and the start of a numerical solution to the differential equations. The first step of the numerical solution provides a new value of x_d at a later time h. These new values are then substituted into the algebraic equations and a new value of x_a determined numerically. The new value of x_a is substituted into the differential equations and the numerical solution continues.

The validity of the non-linear formulation and solution technique has been demonstrated in a recent version of MISSAP by the inclusion of a non-linear transistor model in the component library. The example in the following section contains this non-linear model.

2.4 NON-LINEAR EXAMPLE

Consider the electrical network shown in Figure A-3.



where

$R_1 = 30 \times 10^3$ $\quad R_7 = 10^6$ $\quad A_1$ = Ammeter

$R_2 = 10$ $\quad R_8 = 10^6$ $\quad V_{2-6}$ = Voltmeter

$R_3 = 10^6$ $\quad L_1 = 3 \times 10^3$ $\quad X_i$ L/meter $= .25 \times 10^{-6}$

$R_4 = 10^6$ $\quad C_1 = 1000 \times 10^{-12}$ $\quad$ C/meter $= 100 \times 10^{-12}$

$R_5 = 10 \times 10^3$ $\quad C_2 = .001 \times 10^{-6}$ $\quad$ Length = 1000 meter

$R_6 = 150$ $\quad K_1 = -10^6$ $\quad E_e = +20$

$K_2 = -10^6$ $\quad E_c = +20$

$$E_s = 20\left[1 - e^{-3 \times 10^6 t}\right] - 20\left[1 - e^{-3 \times 10^6 (t - 5 \times 10^6)}\right] \cup \left[t - 5 \times 10^6\right]$$

The transistor T_1 characteristics are given by the following quantities;
Reverse saturation current of the emitter-base junction = 10^{-6} amp.
Reverse saturation current of the collector-base junction = 10^{-6} amp.
Alpha forward = 0.990

The initial voltage of C_1 is assumed to be + 20 volts. All other initial conditions are assumed to be zero.

Figure A-3. Electrical Network for Non-Linear Example

A listing of the data deck prepared for the network shown in Figure A-3 appears in Figure A-4. An explanation of how the MISSAP data deck is prepared is included as Figure A-5 of this Appendix.

The F card specifies the solution interval in real time and indicates the completion of the data. Note that two cards are necessary for the description of transistors and transmission lines.

A Control Data Corporation 3600 digital computer was used to execute MISSAP with this data deck. Computation time was 7 minutes 30 seconds.

The voltages and currents of the network given in Figure A-3 can be approximated accurately by inspection if the following are noted:

1. The damped tank circuit composed of L_1, R_1, and C_1 does not affect the remaining network.

2. The characteristic impedance of the transmission line is $50\,\Omega$. Hence, the initial pulse height to the transmission line is $20 \times \frac{50}{60}$. After the single pulse, the input reflection coefficient is - 2/3.

3. The output reflection coefficient of the transmission line is essentially + 1.

4. The electrical length of the transmission line is

$$\frac{\frac{\text{length}}{1}}{\sqrt{LC}} = 5\ \mu\ \text{sec.}$$

5. The output of the transmission line drives an analog sign inverter.

6. The transistor output is summed with + 20 volts and drives an analog integrator.

The outputs that were computed and plotted by MISSAP are in excellent agreement with the predicted results. These computed outputs are given in Figure A-6. Plot number 1 shows the current as measured in the RLC tank circuit by meter A_1, shown in Figure A-3. Plot number 2 is the voltage measurement as measured by meter V_2. Plots numbered 3 through 6 are the voltages as measured by meter V_3 through V_6 , respectively. The transistor output V_6 indicates the saturation and cutoff present in the non-linear Ebers-Moll transistor model presently stored in the MISSAP component library.

C*	P*	CN* +	CN* −	MISC.*			
L		02	01	3.E-03			
C		02	03	1.E-09	20.		
A	1	03	01	1.0	2.5E-06		
R		02	04	30.E+03			
E	2	04	01	20.	0.	5.E-06	3.E+06
R		05	04	10.			
V	2	05	01	2.0	2.5E-06		
X		05	01	1000.	.25E-06	100.E-12	0.
X	0	6	16	.0000E+00	.0000E+00	.0000E+00	.0000E+00
V	3	6	16	.3000E+01	.2500E-05	.0000E+00	.0000E+00
R		06	07	1.E+06			
R		07	08	1.E+06			
E	0	8	16	-.1000E+07	.0000E+00	.0000E+00	.0000E+00
V	0	7	16	.0000E+00	.0000E+00	.0000E+00	.0000E+00
R		08	09	1.E+04			
T	0	16	09	1.E-06	.9800E+00		
T		11	09	1.E-06	.3900E+02		
R		10	11	.150E+03			
R		11	13	1.E+06			
C		13	14	.001E-06	0.		
E	0	14	16	-.1000E+07	.0000E+00	.0000E+00	.0000E+00
V	0	13	16	.0000E+00	.0000E+00	.0000E+00	.0000E+00
V	4	8	16	.4000E+01	.2500E-05	.0000E+00	.0000E+00
V	5	11	16	.5000E+01	.2500E-05	.0000E+00	.0000E+00
E	5	10	16	-.2000E+02	.0000E+00	.1000E+10	.0000E+00
V	6	14	16	.6000E+01	.2500E-05	.0000E+00	.0000E+00
R		13	15	1.01E+06			
E	5	15	16	.2000E+02	.0000E+00	.1000E+10	.0000E+10
F				40.E-06			

END OF DATA

*KEY

C = COMPONENT
P = PLOT OR TYPE NUMBER
CN = CONNECTION NODES
MISC. = COMPONENT VALUES, LINE LENGTHS, INITIAL CONDITIONS,
PLOTTING INFORMATION, ETC.

Figure A-4. Example Problem Data Listing

```
THE SPECIFICATION CARDS ARE READ WITH THE FORTRAN FORMAT

        (A1,1X,I3,1X,I3,1X,I3,4E12.4)

*****************************************************************************
1 3-5 7-9 11-13 14-25        26-37        38-49        50-61
*****************************************************************************
 TYPE   NODES  DATA 1        DATA 2       DATA 3       DATA 4
*****************************************************************************

E   0   X   Y   A
     VOLTAGE SOURCE CONNECTED BETWEEN NODES X AND Y SUCH THAT
E(T)=A*V(T)       WHERE V(T) IS THE VOLTAGE METERED AS INDICATED BY THE
NEXT CARD IN THE DATA DECK .
EXAMPLE
E   0  10  20  50.0
V      05  06
ALSO E(T)=A*I(T)       WHERE I(T) IS THE CURRENT METERED AS INDICATED BY
THE NEXT CARD IN DATA DECK
EXAMPLE
E   0  01  02  10.0
A      07  04

.............................................................................

E   1   X   Y   A              T1              T2              K
     SINOSOIDAL VOLTAGE SOURCE CONNECTED BETWEEN NODES X AND Y WITH THE
FOLLOWING EQUATION
E(T)=A*(U(T-T1)-U(T-T2))*SINF(2*PI*K*(T-T1))
WHERE PI=3.14159..........
EXAMPLE
E   1  15  07  10.0          1.2           3.75           2.0

.............................................................................

E   2   X   Y   A              T1              T2              K
     EXPONENTIAL VOLTAGE SOURCE CONNECTED BETWEEN NODES X AND Y WITH
THE FOLLOWING EQUATION
E(T)=A*(U(T-T1)(1-EXPF(-K*(T-T1)))-U(T-T2)(1-EXPF(-K*(T-T2))))
EXAMPLE
E   2  00  12  5.0           2.0           3.0           10.0
NOTE - K OF THE ABOVE FORMULA IS A FLOATING POINT NUMBER .

.............................................................................

E   3   X   Y   A              T1              T2              T3
     A TRIANGULAR VOLTAGE SOURCE CONNECTED BETWEEN NODES X AND Y WITH
FOLLOWING EQUATION
E(T)=A*(U(T-T1)*(T-T1)/(T2-T1)-U(T-T2)*(T3-T1)/(T2-T1)*(T-T2)/(T3-T2))
     *(1-U(T-T3))
EXAMPLE
E   3  00  01  10.0          1.0           2.0           4.0

.............................................................................

E   4   X   Y   A              T1              T2
   A SINE SQUARE VOLTAGE SOURCE CONNECTED BETWEEN NODES X AND Y WITH
FOLLOWING EQUATION
E(T)=A*(U(T-T1)-U(T-T2))*(SINF(2*PI*(T-T1)/(T2-T1)))**2
WHERE PI=3.14159..........
EXAMPLE
E   4  01  00  25.0          0.0           3.0

.............................................................................

E   5   X   Y   A              T1              T2
     A RECTANGULAR PULSE VOLTAGE SOURCE CONNECTED TO NODES X AND Y WITH
THE FOLLOWING EQUATION
E(T)=A*(U(T-T1)-U(T-T2))
EXAMPLE
E   5  00  11  3.5           10.0          11.0

       IF T1 AND T2 ARE BOTH ZERO, ABOVE CARD REPRESENTS A UNIT IMPULSE
FUNCTION.
EXAMPLE
E   5  00  21

.............................................................................

E   6   X   Y
     VOLTAGE SOURCE CONNECTED BETWEEN NODES X AND Y WHERE VOLTAGE IS
A FUNCTION OF EXTERNAL DATA APPEARING AFTER THE F CARD WITH THE FOLLOW-
ING FORMAT
    N
             T(1)            X(1)
             T(2)            X(2)
              .               .
              .               .
             T(N)            X(N)
WHERE N DENOTES THE NUMBER OF DATA CARDS AND IS LESS THAN OR EQUAL TO
100 AND WHERE X(I) IS THE VALUE OF VOLTAGE AT TIME T(I) . LINEAR INTER-
POLATION DETERMINES THE VALUES OF X AT POINTS INTERMEDIATE TO THE GIVEN
T VALUES.
EXAMPLE
E   6  01  02
```

Figure A-5. MISSAP Data Deck Preparation Outline (1 of 3)

```
F              5.6
    4
               0.0           1.2
               0.1           1.3
               0.4          -5.4
               1.0           0.0
NOTE - THERE CAN BE AT MOST TWO SUCH DRIVERS IN EACH SET OF DATA .

.........................................................................

FOR CURRENT SOURCES REPLACE LETTER (E) WITH LETTER (I) IN ABOVE CARDS .
THERE CAN BE AT MOST 25 VOLTAGE AND CURRENT SOURCES IN ONE DATA SET.
THERE MAY BE NO VOLTAGE OR CURRENT SOURCES.
NOTICE - DATA1 TO DATA4 MUST ALWAYS BE FLOATING POINT CONSTANTS.

.........................................................................

( IN THE FOLLOWING CARDS LETTER (J) DENOTES A LABEL NUMBER WHICH IS NOT
NEEDED BUT IS ALLOWED FOR IDENTIFICATION PURPOSES.)

.........................................................................

C   J  N1  N2  A               V0
      CAPACITOR WITH CAPACITANCE A AND INITIAL VOLTAGE V0 CONNECTED
BETWEEN NODES N1 AND N2 .
EXAMPLE
C  02  01  02  0.0002          0.0

.........................................................................

R   J  N1  N2  A
      RESISTOR WITH RESISTANCE A CONNECTED BETWEEN NODES N1 AND N2 .
EXAMPLE
R  01  02  05  0.10E+06

.........................................................................

L   J  N1  N2  A               I0
      INDUCTOR WITH INDUCTANCE A AND INITIAL CURRENT I0 CONNECTED BETWEEN
NODES N1 AND N2 .
EXAMPLE
L  04  05  08  0.0001          10.0

.........................................................................

M   J  N1  N2  L1              I01          K21
M   J  N4  N3  L2              I02          K12
      ABOVE TWO CARDS REPRESENT MUTUAL INDUCTANCES WITH ONE INDUCTOR
HAVING TERMINALS N1 AND N2 , SELF INDUCTANCE L1, INITIAL CURRENT I01 AND
COEFICIENT OF COUPLING K21. SECOND INDUCTOR HAS TERMINALS N4 AND N3,
SELF INDUCTANCE L2, INITIAL CURRENT I02 AND COEFFICIENT OF COUPLING
K12.
NOTE - K12 AND K21 MUST BE LESS THAN 0.999 . (FOR PERFECT COUPLERS,
K=1.0 , USE DEPENDENT SOURCES, E O OR I O .)
EXAMPLE
M  01  02  03  0.001           0.0          0.5
M  01  05  07  0.01            1.2          0.75

.........................................................................

X   J  N1  N2  D               L            C            R
X   N  N4  N3  G               L1           R1           R2
      THESE TWO CARDS PLUS THE N CARDS THAT FOLLOW THEM DENOTE A TRANS-
MISSION LINE WITH N METERS PLACED ALONG IT . D IS THE LENGTH OF THE
LINE. L,C,R, AND G ARE THE USUAL PARAMETERS DEFINED PER METER LENGTH.
L1, R1, AND R2 ARE PARAMETERS FOR DISTRIBUTED LINE LOADING SUCH THAT R1
AND L1 ARE CONNECTED IN SERIES AND TOGETHER ARE IN PARALLEL WITH R2. N IS THE
NUMBER OF METERS PLACED ALONG THE TRANSMISSION LINE. CARDS FOR THESE METERS
MUST FOLLOW THE ABOVE TWO CARDS AND MUST HAVE THE FOLLOWING FORMAT .
V   J          X1              X2           X3
      WHERE X1 AND X2 ARE AS EXPLAINED FOR METERS AND X3 DENOTES THE
DISTANCE OF THE METERS FROM THE (N1,N2) END OF THE TRANSMISSION LINE. NOTE THAT
N1 AND N4 MUST BE ON THE SAME LINE.
EXAMPLE
X  06  01  02  100.0           0.01         0.02         100.0
X  03  04  03  0.5             0.1          10.0         12.0
A  01          1.0             0.1          50.0
A  02          0.0             99.0         75.0
V  03          2.0             1.0          100.0
NOTE - NO MORE THAN 6 TRANSMISSION LINES AND NO MORE THAN A TOTAL OF 25
LINE METERS MAY BE SPECIFIED IN ONE SET OF DATA.

.........................................................................

D   J  N1  N2  A               B
      DIODE CONNECTED BETWEEN NODES N1 AND N2 WITH FORWARD DIRECTION FROM
N1 TO N2.
THE EQUATION FOR THE DIODE IS GIVEN BY
               ID=A*(EXPF(B*VD)-1.0)
WHERE ID AND VD ARE THE DIODE CURRENT AND VOLTAGE RESPECTIVELY.
IF THE CARD FIELD CORRESPONDING TO A IS LEFT BLANK,THE VALUE 1.0E-6 WILL
BE USED FOR A.
IF THE CARD FIELD CORRESPONDING TO B IS LEFT BLANK, THE VALUE 39.0 WILL
BE USED FOR B.
EXAMPLE
D  02  03  05  1.0E-8        24.0
```

Figure A-5. MISSAP Data Deck Preparation Outline (2 of 3)

```
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••

T   J  N1  N2    A          B
T   J  N3  N4    C          D
     THE ABOVE TWO CARDS REPRESENT A TRANSISTOR WITH THE FORWARD
DIRECTION OF THE EMITTER-BASE JUNCTION FROM N1 TO N2 AND THE FORWARD
DIRECTION OF COLLECTOR-BASE JUNCTION FROM N3 TO N4. A IS THE REVERSE
SATURATION CURRENT OF THE EMITTER-BASE JUNCTION. C IS THE REVERSE SAT
CURRENT OF THE COLLECTOR-BASE JUNCTION. B IS ALPHA FORWARD AND D IS T
EXPONENTIAL COEFICIENT Q/MKT. ALL OF THE DATA VALUES MUST BE POSITIVE
THE DISTINCTION BETWEEN NPN AND PNP TRANSISTORS IS MADE BY THE ORDERI
OF THE NODE NUMBERS TO CORRESPOND TO THE JUNCTIONS FORWARD DIRECTION.
     THE FOLLOWING EQUATIONS WILL BE SATISFIED

          E=B*A/C          ALPHA INVERSE
          I1=A/(1.0-B*E)
          I2=C/(1.0-B*E)
          IE=I1*(EXPF(D*VE)-1.0)+E*I2*(EXPF(D*VC)-1.0)
          IC=B*I1*(EXPF(D*VE)-1.0)+I2*(EXPF(D*VC)-1.0)

 WHERE IE AND IC ARE THE EMITTER AND COLLECTOR CURRENTS RESPECTIVELY
VE AND VC ARE THE EMITTER-BASE AND COLLECTOR-BASE VOLTAGES RESPECTIVE
ANY OR ALL OF THE DATA FIELDS MAY BE LEFT BLANK IN WHICH CASE THE FOL
VALUES WOULD BE SUBSTITUTED FOR THE BLANK SPECIFICATIONS
          A=1.0E-6
          B=0.98
          C=1.0E-6
          D=39.0
EXAMPLE FOR  PNP
T  01  05  06    1.0E-8       0.99
T  01  05  08    1.0E-7        30.0

NOTE - THE MAXIMUM NUMBER OF TRANSISTORS AND DIODES WHICH MAY BE SPEC
IS LIMITED SUCH THAT. TWICE THE NUMBER OF TRANSISTORS PLUS THE NUMBER
DIODES IS LESS THAN OR EQUAL TO 16.

••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••

V    J  N1  N2  X1              X2              X3
     THIS CARD SPECIFIES A VOLTMETER CONNECTED TO NODES N1 AND N2.
REPLACING (V) WITH (A) DENOTES AN AMMETER.
X1 DETERMINES THE TYPE OF OUTPUT TO BE PLOTTED ON A LINE PRINTER BY
MISSAP. X2 DENOTES THE SCALE FOR SUCH A PLOT AS FOLLOWS
X3 IS A PARAMETER USED WHEN A METER IS PLACED ALONG A TRANSMISSION LI
                  X3 IS ALSO USED AS AN UPPER BOUND FOR INVERSE
FOURIER TRANSFORM PLOTS. (SEE PART B BELOW.)

A ) X1 A POSITIVE INTEGER (IN FLOATING POINT FORMAT) DENOTES THE PLOT
NUMBER AND INSTRUCTS MISSAP TO PRODUCE A TIME PLOT OF THE MEASURED QU
UP TO EIGHT METERS MAY HAVE THE SAME PLOT NUMBER IN WHICH CASE THE QU
THEY MEASURE WILL BE PLOTTED ON THE SAME AXES. X1 MUST BE AN INTEGRAL
SEQUENCE STARTING WITH VALUE  1.0 .E.G. 1.0 , 2.0 , 3.0 , ........
A BREAK IN THE SEQUENCE OF NUMBERING WILL INHIBIT THE PLOTTING OF THO
ABOVE THE BREAK.
X2 IS THE TIME DIVISION USED FOR PLOTTING AND CORRESPONDS TO THE TIME
OF TEN PRINTER PAGE LINES. FOR ALL METERS WHICH HAVE THE SAME X1 VALU
THE LARGEST OF THEIR X2 VALUES WILL BE CHOSEN AS THE TIME DIVISION.
IF ALL METERS WITH THE SAME X1 VALUE MEASURE CONSTANT QUANTITIES THEN
INSTEAD OF THE PLOT THE FOLLOWING WILL BE PRINTED OUT

     'CURVES OF CONSTANT VALUES - PLOT SUPPRESSED.'

B ) X1=0.0 INSTRUCTS MISSAP TO PRODUCE THE FOURIER TRANSFORM OF THE M
QUANTITY. THE MAGNITUDE AND PHASE WILL BE PLOTTED OVER A THREE DECADE
OF FREQUENCY ON LOG-LOG SCALES . THE PLOT WILL BEGIN AT A FREQUENCY T
IS THE FIRST MULTIPLE OF TEN LESS THAN OR EQUAL TO THE RECIPROCAL OF
TIME INTERVAL OVER WHICH THE SOLUTION HAS BEEN OBTAINED. (VALUE ON TH
F CARD.)
X2 DENOTES THE NUMBER OF PRINTER PAGE LINES THAT THE THREE DECADE FRE
SCALE WILL OCCUPY. IF X2 IS LESS THAN 51, THE VALUE OF X2 WILL BE RES
51. IF THE MAXIMUM OF THE MAGNITUDE OF THE FOURIER TRANSFORM IS LESS
0.1E-24 NO PLOT WILL BE PRODUCED TO AVOID POSSIBLE INACCURACIES.
IF X3 IS DIFFERENT FROM ZERO,THE reciprocal of FOURIER TRANSFORM OF THE
QUANTITY METERED WILL BE PLOTTED. VALUE OF X3 DENOTES THIS UPPER BOUN
ALL VALUES LARGER THAN THIS  UPPER BOUND WILL BE PRINTED OUT

C ) A NEGATIVE INTEGER (IN FLOATING POINT FORMAT) IS A REQUEST FOR A
PARAMETRIC PLOT. QUANTITIES MEASURED BY TWO METERS HAVING THE SAME NE
X1 VALUES WILL BE PLOTTED ONE AS A FUNCTION OF THE OTHER. THE PARAMET
WHICH RELATES THE TWO BEING TIME. X2 INDICATES THE TIME INTERVAL REPR
BY TEN CONSEQUTIVE POINTS OF THE PLOT. THE POINT OF THE PLOT CORRESPO
TO TIME 0.0 IS DENOTED BY LETTER (S). EVERY TENTH POINT IS DENOTED BY
A NUMBER FROM 0 TO 9 . THESE NUMBERS INCREASE IN THE DIRECTION OF INC
TIME. AFTER REACHING NUMBER 9 THE SEQUENCE WILL BE REPEATED STARTING
ZERO. AND SO ON.

NOTE - THERE CAN BE AT MOST 25 METERS ASSOCIATED WITH THE LUMPED PARA
PARTS OF THE NETWORK. THIS IS IN ADDITION TO THE 25 METERS ALLOWED IN
TRANSMISSION LINE SYSTEMS.

NOTE - THE MAXIMUM LENGTH OF TIME PLOTS AND FOURIER TRANSFORM PLOTS I
CONFINED TO FIVE PAGES. THIS IS A SAFEGUARD AGAINST UNNECESSARY OUTPU

••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••

F              T
THIS CARD MUST TERMINATE A DATA SET(EXCEPT FOR POSSIBLE TABULAR DATA
ASSOCIATED WITH DRIVERS OF TYPE 6).  THE SOLUTION OF THE NETWORK EQUA
WILL BE OBTAINED FROM TIME EQUAL TO ZERO TO TIME EQUAL TO T.

••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

Figure A-5. MISSAP Data Deck Preparation Ou

•····

URATION
HE
•
NG

AND
LY.
LOWING

IFIED
OF

....

NE.

ANTITY.
ANTITIES

SE

E.

TERED
RANGE
HAT
THE
E

QUENCY
ET TO
THAN

D.

GATIVE
ER
ESENTED
NDING

REASING
WITH

METER
THE

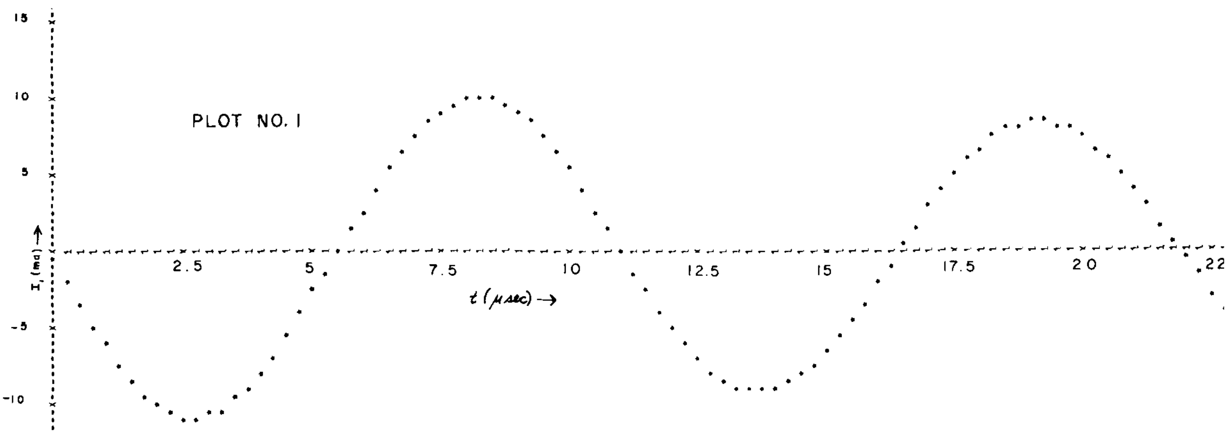
S
T.

....

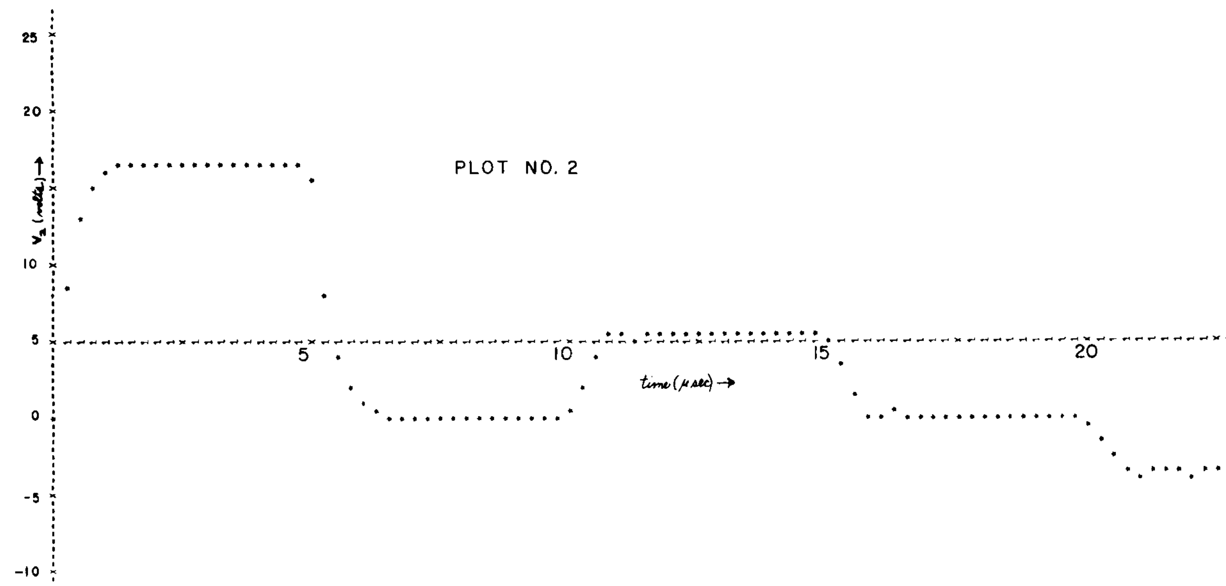
TIONS

....

tline (3 of 3)



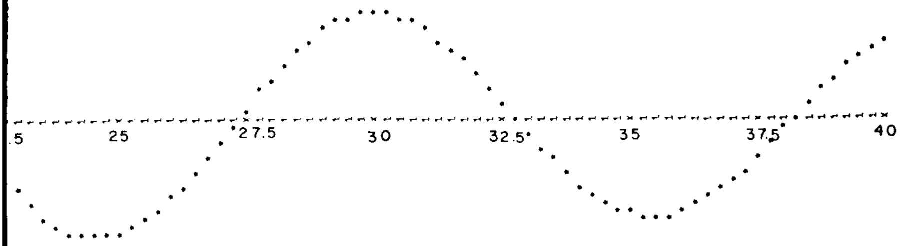
PLOT NO. 1
15
10
5
-5
-10
2.5
5
7.5
10
12.5
15
17.5
20
t (μsec) →

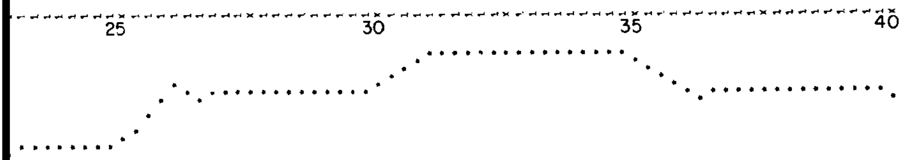


PLOT NO. 2
25
20
10
5
0
-5
-10
5
10
15
20
time (μsec) →

Figure A-6.

#1





Computer Outputs for Non-Linear Example (1 of 3)

#2

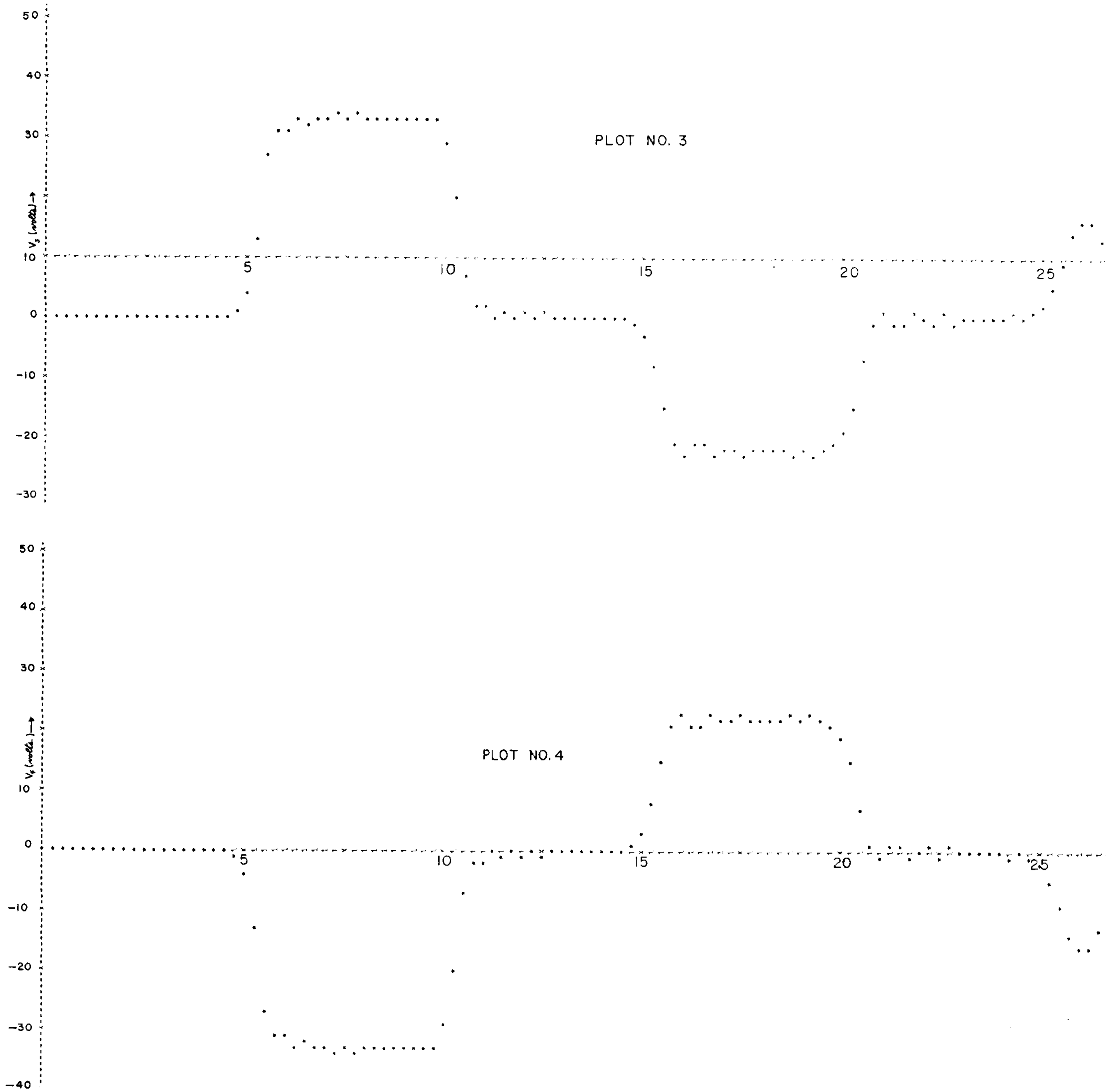


Figure A-6. Computer Outputs for Non-Linear Example (2 of 3)

#1

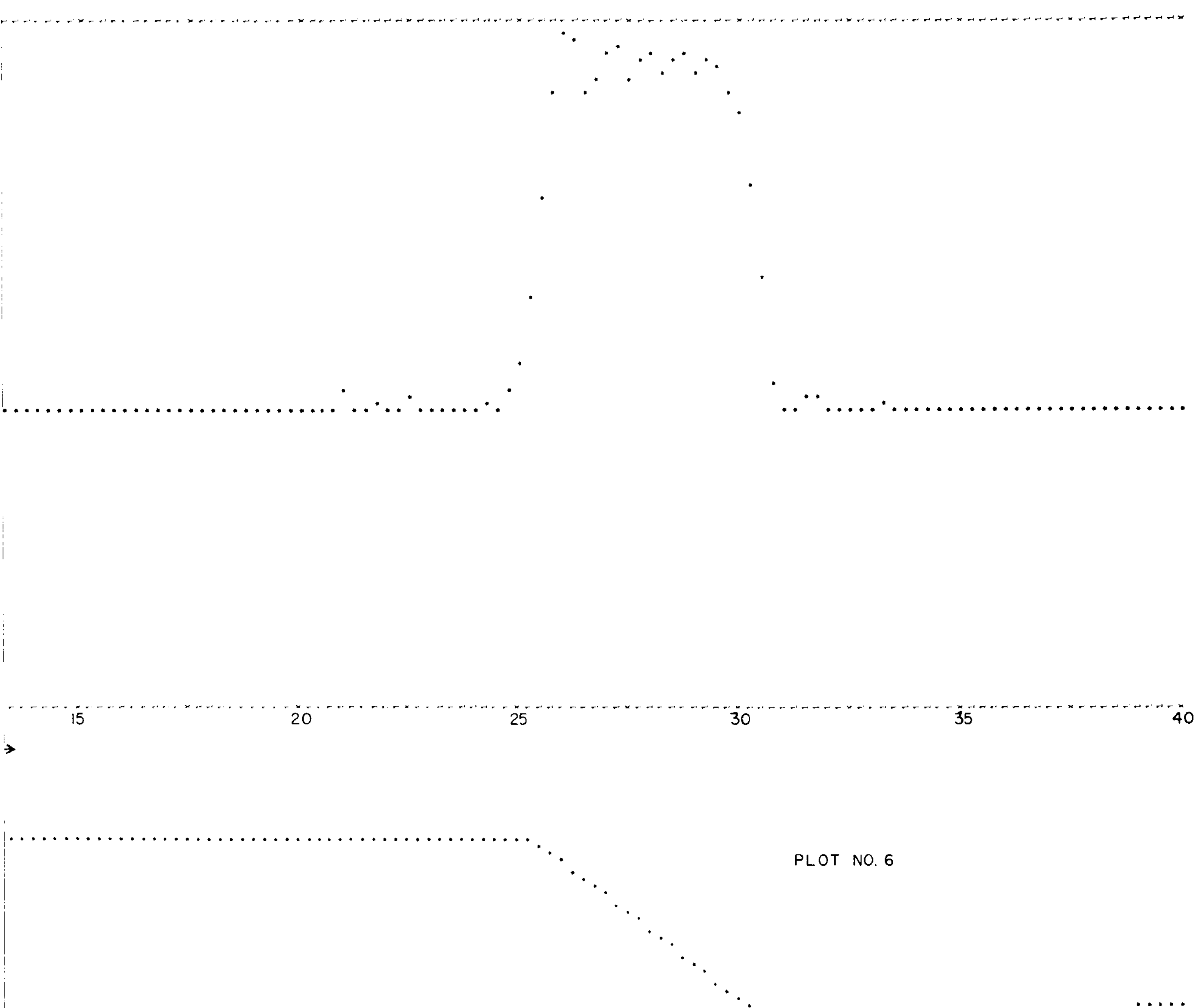
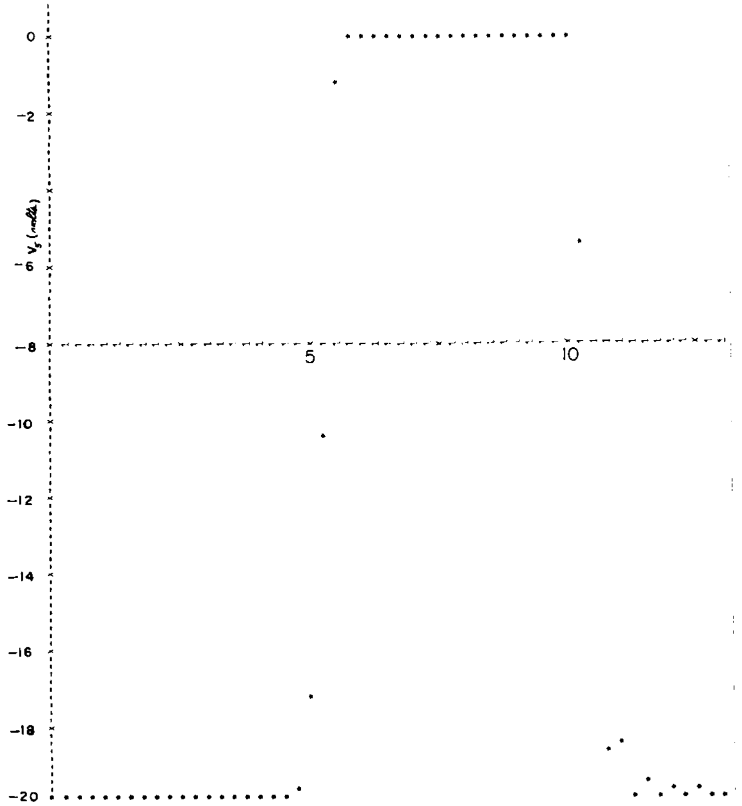


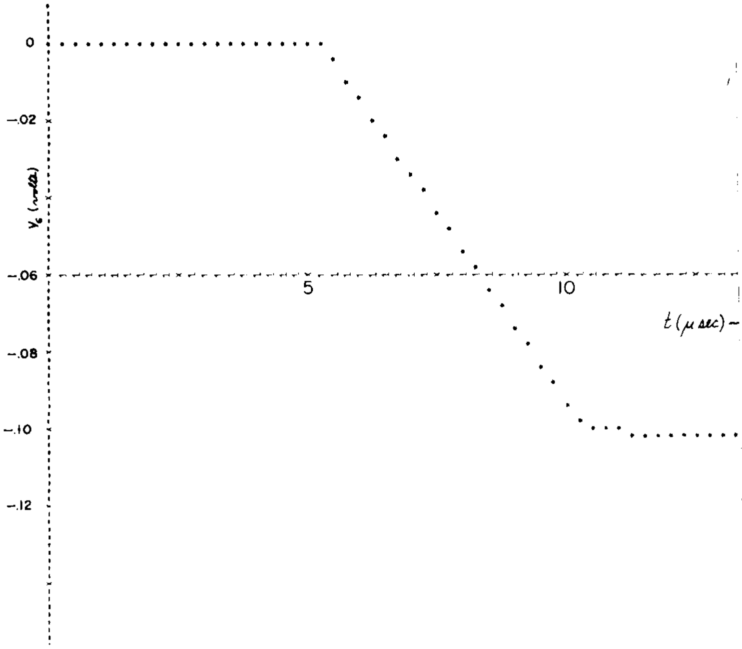
Figure A-6. Computer Outputs for Non-Linear Example (3 of 3)

#2

PLOT NO. 5

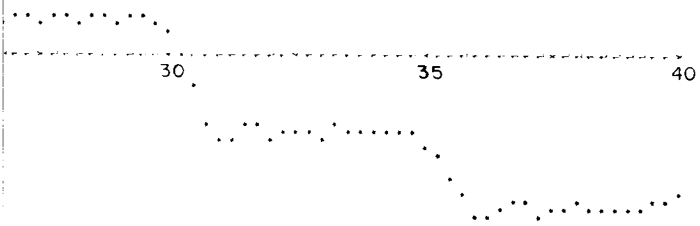


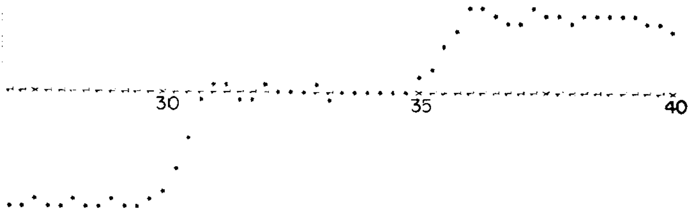
0
-2
V_5 (volts)
-6
-8
5
10
-10
-12
-14
-16
-18
-20



0
-.02
V_6 (volts)
-.06
5
10
t (μ sec) →
-.08
-.10
-.12

#1





#2

3 PERFORMANCE MODEL GENERATION

As discussed in the introduction to this Appendix, it is desirable to construct a performance model such that the system variable frequency distributions may be expressed as functions of the component parameter or coefficient frequency distributions. The problem of obtaining an exact relationship of this type is impossible for all but the simplest cases. However, it is felt that through use of carefully chosen approximation techniques, a suitable solution to the problem is obtainable. The following discussion is, therefore, limited to a means of approximating the functional relationship for small perturbations of the component coefficients about the nominal values. It is felt that the small perturbation assumption is certainly justifiable in studying such things as the accuracy of present day missile systems since the component tolerances normally associated with these systems restrict the coefficients to a very small range.

For convenience, this section will deal primarily with an error analysis type of problem. In general, the performance analysis problem is broader than this. However, the concepts discussed here could conceivably be extended to include the entire performance analysis area.

By first determining the error approximation function, the statistical parameters (moments) of the system error distribution can then be related to the statistical parameters of the components by means of the approximation functions. For example, if the linear terms of an m-dimensional Taylor series (m being the number of independent coefficient errors) is used for the approximation, the output error variance is given by

$$\sigma_{out}^2 = a_1^2 \sigma_1^2 + a_2^2 \sigma_2^2 + \ldots + a_m^2 \sigma_m^2 \qquad \text{Eq. 26}$$

where the a's are the constants determined by the Taylor series expansion. This type of functional relationship between the system output and component statistical parameters is of the desired form for optimization and can be used directly in the PRESTO minimization technique.

The system state model that is explained and formulated in Section 2 of this Appendix, provides the basis for investigating three different methods of generating the system error versus component error approximation function. In all cases, it is assumed that the system model is given as

$$\dot{\bar{x}} = \bar{g}(\bar{x}, \bar{u}) \qquad \text{Eq. 27}$$

$$v = \bar{f}(\bar{x}, \bar{u}) \qquad \text{Eq. 28}$$

where $\bar{u}$ is a column matrix (vector) of specified inputs to the system. Equation 27 represents a system of n simultaneous non-linear differential equations and Equation 28 relates the output (v) of the system (position, velocity, etc.) to the state variable $\bar{x}$ and the input $\bar{u}$.

The following three methods provide a technique for approximating the unknown functional relationship

$$\Delta v\ (t=T) = F(\epsilon_1, \epsilon_2, \ldots \epsilon_m) \qquad \text{Eq. 29}$$

where Δv is a system output error, $\epsilon_1, \epsilon_2, \ldots \epsilon_m$ are small component coefficient errors and T is a terminal time. Since one would expect the system output error to be reasonably well behaved about the nominal trajectory and have the property of being an "almost odd" function of any single component error, i.e.,

$$\Delta v(t=T) = F(0, 0, \ldots, 0, \epsilon_i, 0, \ldots 0) \simeq -F(0, 0, \ldots 0, -\epsilon_i, 0, \ldots 0) \qquad \text{Eq. 30}$$

for ϵ_i small, the most logical first approximation to this function is the sum of m linear terms. The best improvement of the linear approximation seems to be a truncated m-dimensional Taylor series about the nominal trajectory. For this reason, the truncated Taylor series is selected for all three approximation methods that follow.

The mathematical details for the three techniques are provided only for the linear and quadratic terms of the series. The same methodology could, however, be used to generate any number of higher degree terms if they are required.

Since the details are somewhat lengthy, a brief word description is given here for the convenience of the reader of how the linear Taylor series coefficients (sometimes called sensitivity coefficients) are obtained.

> Method I. The differential equations describing how the sensitivity coefficients change as a function of time are derived. These differential equations are then solved simultaneously with the system model given in Equation 27 between time equal to zero and time equal to T. The sensitivity coefficients of the system at time T can be determined algebraically from the solution to the differential equations.
>
> Method II. This method is basically the same as Method I with one exception; a special property of the differential equations is recognized and exploited to reduce the number of simultaneous differential equations that must be solved to obtain the Taylor series coefficients.
>
> Method III. The component coefficients are sequentially perturbed in the system model and the effect of this perturbed coefficient is determined by re-solving the model. The change in output divided by the coefficient perturbation provides an approximation to the sensitivity coefficient.

The mathematical details of these methods follow.

3.1 METHOD I

The state model of the system with vector notation is

$$\dot{x} = g(x, u, \epsilon) \qquad \text{Eq. 31}$$

$$v = f(x, u, \epsilon) \qquad \text{Eq. 32}$$

where x, u and ϵ are vectors with components $x_1, x_2 \ldots x_n$; $u_1, u_2 \ldots u_q$; $\epsilon_1, \epsilon_2 \ldots \epsilon_m$ respectively, and v is the single system output.

At a particular time T, the truncated Taylor series expansion of the output about the nominal solution is:

$$v = v_{nom} + \sum_{L=1}^{m} \left(\frac{\partial v}{\partial \epsilon_l}\right)_{\substack{nom \\ t=T}} \epsilon_l + \sum_{j=1}^{m} \sum_{k=1}^{m} \left(\frac{\partial^2 v}{\partial \epsilon_j \, \partial \epsilon_k}\right)_{\substack{nom \\ t=T}} \epsilon_j \, \epsilon_k .$$

Eq. 33

For physical systems, it is reasonable to assume that $\frac{\partial v}{\partial \epsilon_i}$ and $\frac{\partial^2 v}{\partial \epsilon_i \, \partial \epsilon_j}$ exist and are continuous in a region about the nominal solution. Therefore

$$\left(\frac{\partial^2 v}{\partial \epsilon_j \, \partial \epsilon_k}\right)_{nom} = \left(\frac{\partial^2 v}{\partial \epsilon_k \, \partial \epsilon_j}\right)_{nom} .$$

The problem is, of course, to find the time varying coefficients $\left(\frac{\partial v}{\partial \epsilon_i}\right)_{nom}$ and $\left(\frac{\partial^2 v}{\partial \epsilon_j \, \partial \epsilon_k}\right)_{nom}$ evaluated at T seconds.

To determine the first degree coefficients, differentiate (2) with respect to ϵ_i :

$$\frac{\partial v}{\partial \epsilon_i} = \frac{\partial f(x, u, \epsilon)}{\partial x} \times \frac{\partial x}{\partial \epsilon_i} + \frac{\partial f(x, u, \epsilon)}{\partial \epsilon_i} \qquad \text{Eq. 34}$$

where:

$$\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \; \text{-------} \; \frac{\partial f}{\partial x_n} \right]$$

and

$$\frac{\partial x}{\partial \epsilon_i} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \epsilon_i} \\ \\ \dfrac{\partial x_2}{\partial \epsilon_i} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{\partial x_n}{\partial \epsilon_i} \end{bmatrix}$$

If $\hat{x}$ represents the nominal solution to Equation 31 i.e., $\dot{\hat{x}} = g(\hat{x}, u, 0)$, then

$$\left(\frac{\partial v}{\partial \epsilon_i}\right)_{nom} = \left.\frac{\partial f(x, u, \epsilon)}{\partial x}\right|_{\substack{x=\hat{x}\\ \epsilon=0}} \times \frac{\partial x}{\partial \epsilon_i} + \left.\frac{\partial f(x, u, \epsilon)}{\partial \epsilon_i}\right|_{\substack{x=\hat{x}\\ \epsilon=0}} \qquad \text{Eq. 35}$$

where $\frac{\partial x}{\partial \epsilon_i}$ is the solution to the time-varying linear differential equations obtained by differentiating Equation 31 with respect to ϵ_i.

$$\frac{\partial \dot{x}}{\partial \epsilon_i} = \frac{d}{dt}\left(\frac{\partial x}{\partial \epsilon_i}\right) = \left.\frac{\partial g(x, u, \epsilon)}{\partial x}\right|_{\substack{x=\hat{x}\\ \epsilon=0}} \times \frac{\partial x}{\partial \epsilon_i} + \left.\frac{\partial g(x, u, \epsilon)}{\partial \epsilon_i}\right|_{\substack{x=\hat{x}\\ \epsilon=0}} .$$

Rewriting this in matrix form with evaluation after differentiation implied:

$$\frac{d}{dt}\begin{bmatrix} \frac{\partial x_1}{\partial \epsilon_i} \\ \frac{\partial x_2}{\partial \epsilon_i} \\ \vdots \\ \frac{\partial x_n}{\partial \epsilon_i} \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1(\hat{x},u,0)}{\partial x_1} & \frac{\partial g_1(\hat{x},u,0)}{\partial x_2} & \cdots & \frac{\partial g_1(\hat{x},u,0)}{\partial x_n} \\ \frac{\partial g_2(\hat{x},u,0)}{\partial x_1} & & & \vdots \\ \vdots & & & \vdots \\ \frac{\partial g_n(\hat{x},u,0)}{\partial x_1} & \cdots & \cdots & \frac{\partial g_n(\hat{x},u,0)}{\partial x_n} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial \epsilon_i} \\ \frac{\partial x_2}{\partial \epsilon_i} \\ \vdots \\ \frac{\partial x_n}{\partial \epsilon_i} \end{bmatrix} + \begin{bmatrix} \frac{\partial g_1(\hat{x},u,0)}{\partial \epsilon_i} \\ \frac{\partial g_2(\hat{x},u,0)}{\partial \epsilon_i} \\ \vdots \\ \frac{\partial g_n(\hat{x},u,0)}{\partial \epsilon_i} \end{bmatrix}$$

Eq. 36

Since it is assumed that the initial conditions of Equation 31 are not functions of the component coefficients, the initial conditions associated with Equation 36 are zero.

In general, both the nominal system and Equation 36 must be solved simultaneously between $t = 0$ and $t = T$ by numerical techniques. Note however, that Equation 36 can be solved after the nominal solution is obtained if $\hat{x}$ is stored for later evaluation of $\frac{\partial g}{\partial x}(\hat{x}, u, 0)$ and $\frac{\partial g}{\partial \epsilon_i}(\hat{x}, u, 0)$. The first degree coefficients of Equation 33 are obtained by substituting the solution to Equation 36 at $t = T$ into Equation 35.

The second degree coefficients of Equation 33 are obtained in a similar manner. Consider

$$\frac{\partial^2 v}{\partial \epsilon_k \, \partial \epsilon_j} = \left[\frac{\partial f(\hat{x}, u, 0)}{\partial \epsilon_k \, \partial x} + \sum_{r=1}^{n} \frac{\partial^2 f(\hat{x}, u, 0)}{\partial x_r \, \partial x} \, \frac{\partial x_r}{\partial \epsilon_k} \right] \frac{\partial x}{\partial \epsilon_j}$$

$$+ \frac{\partial f(\hat{x}, u, 0)}{\partial x} \, \frac{\partial^2 x}{\partial \epsilon_k \partial \epsilon_j} + \frac{\partial^2 f(\hat{x}, u, 0)}{\partial \epsilon_k \, \partial \epsilon_j} + \frac{\partial^2 f(\hat{x}, u, 0)}{\partial \epsilon_j \, \partial x} \, \frac{\partial x}{\partial \epsilon_k}$$

Eq. 37

where:

$$\frac{\partial^2 f}{\partial \epsilon_k \partial x} = \left[\frac{\partial^2 f}{\partial \epsilon_k \partial x_1} \quad \frac{\partial^2 f}{\partial \epsilon_k \partial x_2} \quad \text{---------} \quad \frac{\partial^2 f}{\partial \epsilon_k \partial x_n} \right]$$

Eq. 38

$$\frac{\partial^2 f}{\partial x_r \partial x} = \left[\frac{\partial^2 f}{\partial x_r \partial x_1} \quad \frac{\partial^2 f}{\partial x_r \partial x_2} \quad \text{---------} \quad \frac{\partial^2 f}{\partial x_r \partial x_n} \right]$$

Eq. 39

$$\frac{\partial x}{\partial \epsilon_j} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \epsilon_j} \\ \\ \dfrac{\partial x_2}{\partial \epsilon_j} \\ \vdots \\ \dfrac{\partial x_n}{\partial \epsilon_j} \end{bmatrix}$$

Eq. 40

$$\frac{\partial^2 x}{\partial \epsilon_k \; \partial \epsilon_j} = \begin{bmatrix} \dfrac{\partial^2 x_1}{\partial \epsilon_j \, \partial \epsilon_k} \\ \dfrac{\partial^2 x_2}{\partial \epsilon_j \; \partial \epsilon_k} \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{\partial^2 x_n}{\partial \epsilon_j \; \partial \epsilon_k} \end{bmatrix} \qquad \text{Eq. 41}$$

The vector $\dfrac{\partial^2 x}{\partial \epsilon_j \; \partial \epsilon_k}$ is the solution to

$$\frac{d}{dt}\left(\frac{\partial^2 x}{\partial \epsilon_j \; \partial \epsilon_k}\right) = \frac{\partial g(\hat{x}, u, 0)}{\partial x} \; \frac{\partial^2 x}{\partial \epsilon_j \partial \epsilon_k} + \frac{\partial^2 g(\hat{x}, u, 0)}{\partial \epsilon_j \; \partial \epsilon_k} + \frac{\partial^2 g(\hat{x}, u, 0)}{\partial \epsilon_j \; \partial x} \; \frac{\partial x}{\partial \epsilon_k}$$

$$+ \left(\frac{\partial^2 g(\hat{x}, u, 0)}{\partial x \quad \partial \epsilon_k} + \sum_{q=1}^{n} \frac{\partial^2 g(\hat{x}, u, 0)}{\partial x \quad \partial x_q} \; \frac{\partial x_q}{\partial \epsilon_k} \right) \frac{\partial x}{\partial \epsilon_j} \qquad \text{Eq. 42}$$

where:

$$\frac{\partial g(\hat{x}, u, 0)}{\partial x} = \begin{bmatrix} \dfrac{\partial g_1 (\hat{x}, u, 0)}{\partial x_1} & \dfrac{\partial g_1 (\hat{x}, u, 0)}{\partial x_2} & \cdots & \dfrac{\partial g_1 (\hat{x}, u, 0)}{\partial x_n} \\ \dfrac{\partial g_2 (\hat{x}, u, 0)}{\partial x_1} & & & \vdots \\ \vdots & & & \vdots \\ \dfrac{\partial g_n (\hat{x}, u, 0)}{\partial x_1} & \cdots & \cdots & \dfrac{\partial g_n (\hat{x}, u, 0)}{\partial x_n} \end{bmatrix} \qquad \text{Eq. 43}$$

$$\frac{\partial^2 g(\hat{x}, u, 0)}{\partial \epsilon_j \ \partial \epsilon_k} = \begin{bmatrix} \dfrac{\partial^2 g_1 (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial \epsilon_k} \\ \\ \dfrac{\partial^2 g_2 (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial \epsilon_k} \\ \vdots \\ \\ \dfrac{\partial^2 g_n (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial \epsilon_k} \end{bmatrix} \qquad \text{Eq. 44}$$

$$\frac{\partial \ g(\hat{x}, u, 0)}{\partial x \ \partial \epsilon_j} = \frac{\partial \ g(\hat{x}, u, 0)}{\partial \epsilon_j \ \partial x} = \begin{array}{cccc} \dfrac{\partial^2 g \ (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial x_1} & \dfrac{\partial^2 g \ (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial x_2} & ---- & \dfrac{\partial^2 g \ (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial x_n} \\ \dfrac{\partial^2 g_2 (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial x_1} & & & \vdots \\ \vdots & & & \vdots \\ \dfrac{\partial^2 g \ (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial x_1} & \multicolumn{2}{c}{-----------------} & \dfrac{\partial^2 g \ (\hat{x}, u, 0)}{\partial \epsilon_j \ \partial x_n} \end{array}$$

Eq. 45

$$
\frac{\partial^2 g(\hat{x},u,0)}{\partial x\, \partial x_q} = \begin{bmatrix} \frac{\partial^2 g(\hat{x},u,0)}{\partial x_1\, \partial x_q} & \frac{\partial^2 g(\hat{x},u,0)}{\partial x_2\, \partial x_q} & \cdots & \frac{\partial^2 g(\hat{x},u,0)}{\partial x_n\, \partial x_q} \\ \frac{\partial^2 g(\hat{x},u,0)}{\partial x_1\, \partial x_q} & & & \vdots \\ \vdots & & & \vdots \\ \frac{\partial^2 g(\hat{x},u,0)}{\partial x_1\, \partial x_q} & \cdots & \cdots & \frac{\partial^2 g(\hat{x},u,0)}{\partial x_n\, \partial x_q} \end{bmatrix} \qquad \text{Eq. 46}
$$

and

$$
\left. \frac{\partial^2 x}{\partial \epsilon_j\, \partial \epsilon_k} \right|_{t=0} = 0 \qquad \text{Eq. 47}
$$

The solutions to Equations 42 and 36 evaluated at $t = T$ are substituted into Equation 37 to obtain the second degree coefficients.

Notice that Equation 42 is a set of time varying linear differential equations with the same homogenous part as Equation 36. The vectors $\frac{\partial x}{\partial \epsilon_j}$ and $\frac{\partial x}{\partial \epsilon_k}$ in Equation 42 are solutions to Equation 36 with $i = j$ and $i = k$ respectively.

It is interesting to compute the number of differential equations that must be solved for a tenth order system $(n = 10)$ with fifty coefficient perturbations $(m = 50)$. In such a case there are:

$m = 50$ first degree terms in the Taylor series.

$\binom{m}{1} + \binom{m}{2} = 1,275$	second degree terms in the Taylor series.
$n = 10$	simulation non-linear differential equations to obtain $\hat{x}$.
$nm = 500$	linear time varying differential equations to obtain $\frac{\partial x}{\partial \epsilon_i}$, $i = 1, 2, ---m$. These may be solved in groups of 10 simultaneous equations with $\hat{x}$ required in each group.
$n \left[\binom{m}{1} \; \binom{m}{2}\right] = 12,750$	linear time varying differential equations to obtain $\frac{\partial^2 x}{\partial \epsilon_j \, \partial \epsilon_k}$, $i=1, 2, ---m$, $j=1, 2, ---m$. These may be solved in groups of 10 simultaneous equations with $\hat{x}$, $\frac{\partial x}{\partial \epsilon_j}$ and $\frac{\partial x}{\partial \epsilon_k}$ required in each group.

3.2 METHOD II

It was observed, for Method I, that the differential equations in $\frac{\partial x}{\partial \epsilon_i}$ and $\frac{\partial^2 x}{\partial \epsilon_j \, \partial \epsilon_k}$ have the same homogeneous part. Marshall[15] has described a technique that exploits this property and, in many cases, reduces the number of differential equations that must be solved. The mechanics of this method are as follows.

Introduce the adjoint system of equations

$$\frac{d}{dt} \Lambda(t) = - \Lambda(t) \frac{\partial g(\hat{x}, u, 0)}{\partial x} \qquad \text{Eq. 48}$$

where $\Lambda(t)$ is an n x n matrix satisfying Equation 48 and $\Lambda(t = 0) = U$ = unit matrix.

The partial derivatives of the state variables with respect to the perturbations are determined as:

$$\begin{bmatrix} \frac{\partial x_1}{\partial \epsilon_1} & \frac{\partial x_1}{\partial \epsilon_2} & \cdots & \frac{\partial x_1}{\partial \epsilon_m} \\ \frac{\partial x_2}{\partial \epsilon_1} & & & \vdots \\ \vdots & & & \vdots \\ \frac{\partial x_n}{\partial \epsilon_1} & & & \frac{\partial x_n}{\partial \epsilon_m} \end{bmatrix}_{t=T} = \Lambda^{-1}(T) \int_0^T \Lambda(t) \begin{bmatrix} \frac{\partial g_1(\hat{x}, u, 0)}{\partial \epsilon_1} & \frac{\partial g_1(\hat{x}, u, 0)}{\partial \epsilon_2} & \cdots & \frac{\partial g_1(\hat{x}, u, 0)}{\partial \epsilon_m} \\ \frac{\partial g(\hat{x}, u, 0)}{\partial \epsilon_1} & & & \vdots \\ \vdots & & & \vdots \\ \frac{\partial g_n(\hat{x}, u, 0)}{\partial \epsilon_1} & \cdots & \cdots & \frac{\partial g_n(\hat{x}, u, 0)}{\partial \epsilon_m} \end{bmatrix} dt$$

Eq. 49

and

$$
\begin{bmatrix} \dfrac{\partial x_1}{\partial \epsilon_j\, \partial \epsilon_k} \\[2ex] \dfrac{\partial x_2}{\partial \epsilon_j\, \partial \epsilon_k} \\ \vdots \\ \dfrac{\partial x_n}{\partial \epsilon_j\, \partial \epsilon_k} \end{bmatrix}_{t=T}
= \Lambda\,(T)^{-1} \int_0^T \Lambda\,(t) \left\{ \begin{bmatrix} \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{\partial \epsilon_j\;\partial \epsilon_k} \\[2ex] \dfrac{\partial^2 g_2\,(\hat{x},u,0)}{\partial \epsilon_j\;\partial \epsilon_k} \\ \vdots \\ \dfrac{\partial^2 g_n\,(\hat{x},u,0)}{\partial \epsilon_j\;\partial \epsilon_k} \end{bmatrix}
+ \begin{bmatrix} \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{\partial \epsilon_j\,\partial x_1} & \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{\partial \epsilon_j\,\partial x_2} & \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{\partial \epsilon_j\,\partial x_n} \\[2ex] \dfrac{\partial^2 g_2\,(\hat{x},u,0)}{\partial \epsilon_j\;\partial x_1} & & \vdots \\ \vdots & & \vdots \\ \dfrac{\partial^2 g_n\,(\hat{x},u,0)}{\partial \epsilon_j\,\partial x_1} & \cdots\cdots & \dfrac{\partial^2 g_n\,(\hat{x},u,0)}{\partial \epsilon_j\;\partial x_n} \end{bmatrix}
\begin{bmatrix} \dfrac{\partial x_1}{\partial \epsilon_k} \\[2ex] \dfrac{\partial x_2}{\partial \epsilon_k} \\ \vdots \\ \dfrac{\partial x_n}{\partial \epsilon_k} \end{bmatrix} +
$$

$$
\left(\begin{bmatrix} \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{\partial x_1\;\partial \epsilon_k} & \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{\partial x_2\;\partial \epsilon_k} & \cdots\; \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{\partial x_n\,\partial \epsilon_k} \\[2ex] \dfrac{\partial^2 g_2\,(\hat{x},u,0)}{\partial x_1\;\partial \epsilon_k} & & \vdots \\ \vdots & & \vdots \\ \dfrac{\partial^2 g_n\,(\hat{x},u,0)}{\partial x_1\;\partial \epsilon_k} & \cdots\cdots & \dfrac{\partial^2 g_n\,(\hat{x},u,0)}{\partial x_n\partial \epsilon_k} \end{bmatrix}
+ \sum_{q=1}^{n} \frac{\partial x_q}{\partial \epsilon_k} \begin{bmatrix} \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{x_1\;\;x_q} & \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{x_2\;\;x_q} & \cdots\; \dfrac{\partial^2 g_1\,(\hat{x},u,0)}{x_n\;x_q} \\[2ex] \dfrac{\partial^2 g_2\,(\hat{x},u,0)}{\partial x_1\;\;\partial x_q} & & \vdots \\ \vdots & & \vdots \\ \dfrac{\partial^2 g_n\,(\hat{x},u,0)}{\partial x_1\;\;\partial x_q} & \cdots\cdots & \dfrac{\partial^2 g_n\,(\hat{x},u,0)}{\partial x_n\;\partial x_q} \end{bmatrix} \right)
\begin{bmatrix} \dfrac{\partial x_1}{\epsilon_j} \\[2ex] \dfrac{\partial x_2}{\partial \epsilon_j} \\ \vdots \\ \dfrac{\partial x_n}{\partial \epsilon_j} \end{bmatrix} \Bigg\}
$$

Eq. 50

The Taylor series coefficients are determined by substituting Equation 49 and 50 into Equations 35 and 37.

An equation count with n = 10 and m = 50 provides:

n = 10	non-linear differential equations to obtain $\hat{x}$.
n = 100	linear time varying differential equations to obtain Λ (t). These may be solved simultaneously in groups of 10 with $\hat{x}$ required in each group.
nm = 500	quadrature evaluations to obtain $\frac{\partial x}{\partial \epsilon_i}$. The matrix Λ (t) and $\hat{x}$ are required for each evaluation.
$n\left[\binom{m}{1}+\binom{m}{2}\right] = 12{,}750$	quadrature evaluations to obtain $\frac{\partial^2 x}{\partial \epsilon_j \partial \epsilon_k}$. The matrix Λ (t), $\hat{x}$, $\frac{\partial x}{\partial \epsilon_j}$ and $\frac{\partial x}{\partial \epsilon_k}$ are required for each evaluation.

3.3 METHOD III

A straightforward approach to the problem of obtaining the first and second degree terms of the Taylor series of Equation 33 is to actually perturb the component coefficients in the nominal equations and observe their effect on the output.

To simplify notation, the output will be considered as only a function of an input vector u and m component perturbations ϵ_1, ϵ_2 ---- ϵ_m, i.e.

$$v = H(\epsilon_1, \epsilon_2, \epsilon_3, \text{----} \epsilon_m, u) \text{ with } (v)_{nom} = H(0, 0, \text{---} 0, u) \qquad \text{Eq. 51}$$

For a sufficiently small increment in ϵ_i , say δ_i , the first and second degree Taylor series coefficients can be approximated as

$$\left.\frac{\partial v}{\partial \epsilon_i}\right|_{\epsilon=0} \doteq \frac{H(0,0,0\text{--}0,\delta_i\ ,0\text{--}0,u) - H(0,0,0\text{--}0,\ -\delta_i\ ,0\text{--}0,u)}{2\delta_i}$$

Eq. 52

$$\left.\frac{\partial^2 v}{\partial \epsilon_i^{\ 2}}\right|_{\epsilon=0} \doteq \frac{1}{\delta_i}\left[\frac{H(0,0,0\text{--}0,\delta_i\ ,0\text{--}0,u) - H(0,0,0\text{---}0,u)}{\delta_i} - \frac{(H(0,0,0\text{---}0,u) - H(0,0,0\text{--}0,-\delta_i\ ,0\text{--}0,u)}{\delta_i}\right]$$

$$= \frac{1}{\delta_i^{\ 2}}\left[H(0,0,0\text{--}0,\delta_i\ ,0\text{--}0,u) - 2H(0,0,0\text{---}0,u) + H(0,0,0\text{--}0,\ -\delta_i,0\text{--}0,u)\right]$$

Eq. 53

$$\underset{j\neq k}{\frac{\partial^2 v}{\partial \epsilon_j\,\partial \epsilon_k}} \doteq \frac{1}{\delta_j}\left[\frac{H(0,0\text{--}0,\ \delta_j\ ,0,\text{--}0,\ \delta_k\ ,0\text{--}0,u) - H(0,0,0\text{--}0,\ \delta_j,0\text{--}0,u)}{\delta_k} - \frac{(H(0,0,0\text{--}0,\ \delta_k,0\text{--}0,u) - H(0,0\text{---}0,u)}{\delta_k}\right]$$

Eq. 54

If, however, the δ selected is too small, the above equations become extremely sensitive to numerical inaccuracies. Therefore to select δ , a rough approximation of the sensitivity of the output to a particular perturbation must be known prior to solving these equations for the Taylor series coefficients.

For $m = 50$ and $n = 10$, the solution requires solving:

$n = 10$	non-linear differential equations for the nominal solution,
$2\,nm = 1000$	non-linear differential equations to determine $\frac{\partial v}{\partial \epsilon_i}$ and $\frac{\partial^2 v}{\partial \epsilon_i^2}$, and
$n \binom{m}{2} = 12{,}250$	non-linear differential equations to determine $\frac{\partial^2 v}{\partial \epsilon_i \, \partial \epsilon_j}$ for $i \neq j$.

These equations may be solved in groups of 10 simultaneous equations.

The equation count in Methods I, II, and III indicates the rather staggering amount of calculations necessary to obtain only the first and second degree coefficients of the Taylor series when the system is moderately large. Higher degree terms require an unreasonable amount of calculation time and are not considered worthwhile for this reason. The methods described in this report could, however, be extended to cover higher degree terms if the increased computation time could be tolerated.

The machine logic associated with the Methods I and II of this section are considerably more complicated than that of III. This additional logic does, however, reduce most of the non-linear differential equations that are present in Method III to linear time varying differential equations. Since the latter is, in general, just as difficult to solve numerically as the non-linear case, Method III appears superior to both I and II. It may be possible to substantially reduce the logic required in Methods I and II by analytical differentiation at the component level and forming the differential equations given in Equations 36, 42, 48, 49, and 50 with a computer.

REFERENCES

1. SELFRIDGE, R. G.: "Coding a General-Purpose Digital Computer to Operate as a Differential Analyzer, "Proc. 1955 Western Joint Computer Conference (IRE).

2. LESH, F.: "Methods of Simulating a Differential Analyzer on a Digital Computer, "J. of the ACM, Volume 5, number 3, 1958.

3. STEIN, M. L., ROSE, J., and PARKER D. B.: "A Compiler with an Analog-Oriented Input Language, "Proc. 1959 Western Joint Computer Conference.

4. HURLEY, J. R.: "DEPI 4 (Differential Equations Pseudo-Code Interpreter)-An Analog Computer Simulator for the IBM 704," Internal Memorandum, Allis Chalmers Mfg., Co., January 6, 1960.

5. HURLEY, J. R.: "Digital Simulation I: DYSAC, A Digitally Simulated Analog Computer," AIEE Summer General Meeting, Denver, Colorado, June 17-22, 1962.

6. STOVER, R. F., and KNUDTSON, H. A.: "H-800 PARTNER-Proof of Analog Results Through a Numerical Equivalent Routine," Doc. No. U-ED 15002, Minneapolis-Honeywell Regulator Co., Aero. Divn., April 30, 1962.

7. GASKILL, R. A., HARRIS, J. W., and MC KNIGHT, A. L.: "DAS- A Digital Analog Simulator," Proc. 1963 Spring Joint Computer Conference.

8. BYRNE, E. R.: "JANIS, A Program for a General-Purpose Digital Computer to Perform Analog Type Simulations," ACM Nat'l. Conf., Denver, Colorado, August 27, 1963.

9. HARNETT, R. T., SANSOM, F. J., and WARSHAWSKY, L. M.: "MIDAS Programming Guide," Tech. Doc. Report SEG-TDR-64-1, Analog Computation Divn., Wright-Patterson Air Force Base, Jan., 1964.

10. BRENNAN, R. D. and SANO, H.: "PACTOLUS - A Digital Simulator Program for the IBM 1620," 1964 Fall Joint Computer Conf., AFIPS Conference Proc., Vol. 26, October 1964.

11. SESHU, S. and REED, M. B.: "Linear Graphs and Electrical Networks," Addison-Wesley, 1961.

12. KOENIG, H. E., and BLACKWELL, W. A.: "Electromechanical System Theory," McGraw-Hill, 1961.

13. WIRTH, J. L.: "Time Domain Models of Physical Systems and Existence of Solutions," Ph. D. Thesis, Michigan State University, 1962.

14. REID, R. J.: "A System Analysis Program," Technical Report No. 1 to IBM, Div. of Engineering Research, Michigan State University, 1962.

15. MARSHALL, W. C.: "Calculus of Perturbations Applied to Lunar Mission Analysis," Progress in Astronautics and Rocketry, Volume 8, pp. 147-194.

APPENDIX B

RELIABILITY

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont)

1 INTRODUCTION

The increasing complexity and cost of systems places an ever greater burden on systems analysis. Analytical methods for conducting such an analysis have been weak in many areas. The PRESTO concept of simulation and optimization is a step toward improving systems analysis techniques.

System reliability is a major portion of the systems analysis problem. Since analytical methods have been either inadequate or impractical or both, the simulation and analysis of system reliability has been more an art than a science. However, increasing importance is being placed on system reliability. Therefore, in developing PRESTO major attention has been devoted to system reliability.

The focal point of the reliability study was the development and application of a Reliability Simulator. This simulator is an integral part of the PRESTO concept. As in other areas of simulation within the framework of the PRESTO concept, the modern high speed digital computer has been utilized in simulation to effect mathematical model generation. This appendix contains a description of the mathematical basis for the Reliability Simulator, with the basis being derived from a system reliability analysis technique known as RAPID*. Two computerized versions of the simulator are also presented.

---

* Reliability Analysis and Prediction Independent of Distributions.

2 RAPID TECHNIQUE

Functionally, RAPID may be considered to involve two distinct phases:

- A Simulation Phase, and
- An Analysis Phase.

The simulation phase requires a symbolic representation of the system under study. This representation consists of defining the system as being composed of a finite number of physically or functionally connected elements. These elements individually may be of different types such as electrical, mechanical, hydraulic, or pneumatic. For each of these elements, a set of element modes is described. The modes of failure are also described for the system. These descriptions serve as inputs to the simulation phase.

The inputs are analyzed to generate a probabilistic model for the system, with the model taking the form of an algebraic equation which can be evaluated to determine the probability of the defined system mode (failure) occurring.

The analysis phase of RAPID utilizes numerical data concerning the probability of the defined element modes occurring as inputs, along with the probabilistic model generated during the simulation phase. The model is then evaluated and the probability of the system failing in the manner defined is calculated. Reliability is equal to one minus this probability.

One significant feature of the RAPID technique is the fact that the entire simulation phase, and hence the model generation, is completely independent of the source of numerical data. Therefore, the numerical data can be changed without generating a new model for the system. This also permits a different failure distribution function to be used to describe each element, if required.

Another feature of the RAPID technique is the ability to consider systems with standby or parallel redundancy. This provides a distinct advantage over the "summing failure rate" technique. Various changes in environment can also be incorporated into the analysis as well as consideration of non-catastrophic failures.

2.1 RAPID SIMULATION PHASE

This section describes the phase of the RAPID technique in which a probabilistic model for the reliability of a system under consideration is developed.

2.1.1 System Description

The following network has been chosen to illustrate the technique and to define some of the nomenclature used in RAPID.

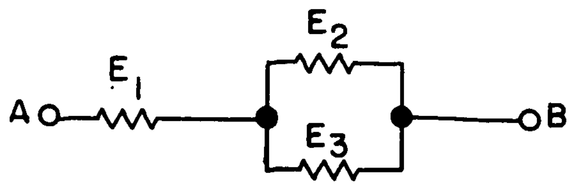


Figure B-1. Basic Example Network

In general, a system is considered to be composed of N physically or functionally connected elements. For the system shown, N is three and the elements are defined to be identical resistors denoted by E_1, E_2, and E_3.

The system is then described by identifying and specifying the allowed element modes for each element in the system.

2.1.1.1 Element Mode

The modes of an element are defined to be a set of mutually exclusive states available to the element. This set is all inclusive in that the element must always exist in one of the defined modes. The set of element modes for the ith element can be represented symbolically by 0_i, 1_i, ..., K_i. This represents the K+1 element modes available to element i.

The probability that element i exists in its K_ith element mode can be represented symbolically by $P(K_i)$, or alternatively by $P(i, K)$. The latter form will be used in this report.

By designating an element mode, K_i, it is understood that the element i also has element modes $0_i, 1_i$, etc. up to and including K_i available to it. Since the element modes for a given element are an all inclusive set of mutually exclusive conditions available to that element, it follows that

$$\sum_{k_i=0}^{K} P(i,k) = 1.$$

The logical union of element modes $1_i, 2_i, \ldots, K_i$ is represented by the symbol $0'_i$. Therefore, by definition

$$0'_i = 1_i \cup 2 \cup \ldots \cup K_i.^*$$

Since the element modes are mutually exclusive, it follows directly from the definitions that

$$P(i,0') = P(1_i \cup 2 \cup \ldots \cup K)$$

$$= P(i,1)+P(i,2)+\ldots+P(i,K).$$

Therefore,

$$P(i,0)+ \sum_{k_i=1}^{K} P(i,k) = 1,$$

or

$$P(i,0)+P(i,0') = 1.$$

The logical union of element modes 0_i and $0'_i$ is represented symbolically by $\underline{\Delta}_i$, and is referred to as the supermode of element i.

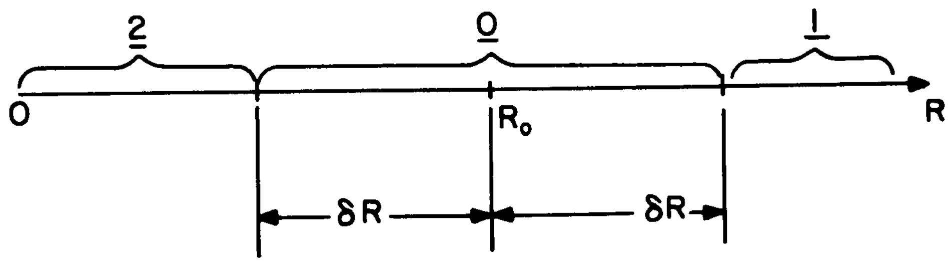
$$P(i,0)+P(i,0') = P(0_i \cup 0') = P(i,\underline{\Delta}) = 1.$$

*The symbols "$\cup$" and "$\cap$" signify the logical union and intersection of events respectively.

For the system under consideration (Figure B-1), the element modes for each element (resistor) are defined as follows:

Element Mode	Element Mode Name	Element Mode Description
$\underline{0}$	GOOD	$R_0 - \delta R \leq R \leq R_0 + \delta R$
$\underline{1}$	OPEN	$R > R_0 + \delta R$
$\underline{2}$	SHORT	$R < R_0 - \delta R$

The underline is used here to denote the mode number as a symbol and not a numeric value. This can be shown graphically in the following diagram.



Only one set of definitions is shown, since all three elements are identical in this example.

2.1.1.2 System States

A system state is identified by defining, for each element in the system, the specific element mode in which the element exists at a given time. For the system under consideration, one such system state is identified by defining each resistor to be open. This system state would be represented symbolically by

$$\underline{1}_1\underline{1}_2\underline{1}_3.$$

The total number of system states, S, for a system of N elements is given by

$$S = \prod_{i=1}^{N} (K_i+1) = (K_1+1)x(K_2+1)x\ldots x(K_N+1).$$

In the formula above, the symbol K_i is treated as a numeric value. K_i can be thought of as the "highest numbered element mode" for the element i, as indicated in Section 2.1.1.1 above.

For the system of three resistors (Figure B-1), the total number of system states, S, is equal to

$$\prod_{i=1}^{3} (K_i+1) = (2+1)x(2+1)x(2+1) = 27.$$

These system states can be listed symbolically as follows:

$\underline{0}_1$	$\underline{0}_2$	$\underline{0}_3$
$\underline{0}_1$	$\underline{0}_2$	$\underline{1}_3$
$\underline{0}_1$	$\underline{0}_2$	$\underline{2}_3$
$\underline{0}_1$	$\underline{1}_2$	$\underline{0}_3$
$\underline{0}_1$	$\underline{1}_2$	$\underline{1}_3$
$\underline{0}_1$	$\underline{1}_2$	$\underline{2}_3$
$\underline{0}_1$	$\underline{2}_2$	$\underline{0}_3$
$\underline{0}_1$	$\underline{2}_2$	$\underline{1}_3$
$\underline{0}_1$	$\underline{2}_2$	$\underline{2}_3$
$\underline{1}_1$	$\underline{0}_2$	$\underline{0}_3$
$\underline{1}_1$	$\underline{0}_2$	$\underline{1}_3$
$\underline{1}_1$	$\underline{0}_2$	$\underline{2}_3$
$\underline{1}_1$	$\underline{1}_2$	$\underline{0}_3$
$\underline{1}_1$	$\underline{1}_2$	$\underline{1}_3$
$\underline{1}_1$	$\underline{1}_2$	$\underline{2}_3$

$\underline{1}_1$	$\underline{2}_2$	$\underline{0}_3$
$\underline{1}_1$	$\underline{2}_2$	$\underline{1}_3$
$\underline{1}_1$	$\underline{2}_2$	$\underline{2}_3$
$\underline{2}_1$	$\underline{0}_2$	$\underline{0}_3$
$\underline{2}_1$	$\underline{0}_2$	$\underline{1}_3$
$\underline{2}_1$	$\underline{0}_2$	$\underline{2}_3$
$\underline{2}_1$	$\underline{1}_2$	$\underline{0}_3$
$\underline{2}_1$	$\underline{1}_2$	$\underline{1}_3$
$\underline{2}_1$	$\underline{1}_2$	$\underline{2}_3$
$\underline{2}_1$	$\underline{2}_2$	$\underline{0}_3$
$\underline{2}_1$	$\underline{2}_2$	$\underline{1}_3$
$\underline{2}_1$	$\underline{2}_2$	$\underline{2}_3$

Identifying all the possible system states for a given system, completely defines the system for the RAPID technique.

2.1.1.3 System Definition Array

Listing all the possible system states for a system of any reasonable size is a formidable task, yet the system must be defined for the simulation phase of the RAPID technique. This same information is conveyed to RAPID in a shorthand notation, by means of the System Definition Array (SDA).

The SDA is a single row array containing N entries for a system of N elements. Symbolically, the SDA is represented as follows:

$$\underline{K}_1 \underline{K}_2 \underline{K}_3 \ldots \underline{K}_N .$$

Each entry represents the "highest numbered" element mode for its corresponding element. The first entry corresponds to element one, the second to element two and so on.

For the example system of three resistors, the System Definition Array is

$$\underline{K}_1 \underline{K}_2 \underline{K}_3 ,$$

or

$$\underline{2}_1\,\underline{2}_2\,\underline{2}_3\,.$$

The subscripts can be dropped for simplicity since the subscript of an entry is identical to its position in the array. The underline can also be omitted, keeping in mind that the entry is a symbol only. Thus, the SDA for the example system would be written as follows:

2 2 2.

2.1.2 System Mode Description

In addition to the System Definition Array, the RAPID technique requires a definition of the system mode under consideration. In effect, this consists of symbolically representing those system states which constitute that particular mode of the system.

In the case of the example system, the system mode to be simulated will be defined as an open circuit between terminals A and B. This condition is represented symbolically by the following eleven system states;

0_1	1_2	1_3
1_1	0_2	0_3
1_1	0_2	1_3
1_1	0_2	2_3
1_1	1_2	0_3
1_1	1_2	1_3
1_1	1_2	2_3
1_1	2_2	0_3
1_1	2_2	1_3
1_1	2_2	2_3
2_1	1_2	1_3

2.1.2.1 System Submodes

The representation of one or more system states in a single row or sequence of element mode symbols is called a system submode.

A shorthand notation is utilized to convey all the system states which represent this system mode under consideration to the simulation phase of the RAPID technique.

In the example system, nine of the system states in the above listing can be represented symbolically with a single row by the use of the supermode Δ. This single row (submode) would be as follows:

$$1_1 \Delta_2 \Delta_3.$$

Associated with each system submode is the order of the submode. The order of a system submode can be determined by inspection by counting the elements that are not represented by a supermode symbol. The system submode

$$1_1 \Delta_2 \Delta_3$$

would therefore be of order one, or a first order submode. A system submode of order N for a system of N elements represents only one system state. Submodes of order less than N represent more than one system state.

2.1.2.2 System Mode Array

The System Mode Array (SMA) is a collection of submodes which are sufficient to describe the entire system mode under consideration. Only those system states which make up the system mode may be included in the array.

By utilizing the element supermode symbol, the system states listed in Section 2.1.2 above, for the open system mode, can be represented by the following SMA:

$$0_1\ 1_2\ 1_3$$

$$1_1\ \Delta_2\ \Delta_3$$

$$2_1\ 1_2\ 1_3.$$

This SMA can be further reduced to two submodes by representing the first and third rows by a single submode;

$$\Delta_1 1_2 1_3.$$

This submode, however, contains one system state which also appears in the second submode of the above SMA. That state is

$$1_1 \; 1_2 \; 1_3.$$

The SMA for the open system mode is thus reduced to two submodes (dropping the subscripts as before),

$$\begin{array}{ccc} 1 & \Delta & \Delta \\ \Delta & 1 & 1 \end{array}$$

having a common system state, or "overlap," between them. This SMA is easily written from an inspection of the system (Figure B-1), by recognizing that

a. The system will be open whenever element E_1 is open, regardless of the condition of element E_2 or element E_3.

b. The system will be open whenever element E_2 <u>and</u> element E_3 are open, regardless of the condition of element E_1.

These two statements lead to the two submodes of the SMA, respectively.

The listing in Section 2.1.2 of all the possible open system states for the example system, made the overlap in the final minimum shorthand representation of the SMA easily detectable. For large systems comprised of elements having numerous modes, however, the listing of all system states for the system mode under study is not practical, because of the magnitude of the listing. The shorthand SMA representation must therefore be allowed without regard to any possible overlap involving one or more system states, yet the final system mode representation for constructing a mathematical model must be a collection of mutually exclusive system submodes. This difficulty is overcome by application of the RAPID STRIP* process to the SMA, so that mutually exclusive

---

*Sorting Technique for the Removal of Intersecting Probabilities system submodes are assured in the mathematical model of the system mode.

2.2 STRIP PROCESS

This process can best be explained by means of an example. Consider the SMA for the open mode of the three resistor systems

$$\begin{matrix} 1 & \Delta & \Delta \\ \Delta & 1 & 1. \end{matrix}$$

Let all the system states be represented symbolically by

$$(\Delta \ \Delta \ \Delta)$$

and referred to as the available system states. Note that this is equivalent to the SDA. The logical intersection of the first system submode and the available system states is taken. This yields

$$(1 \ \Delta \ \Delta) \cap (\Delta \ \Delta \ \Delta) = (1 \ \Delta \ \Delta).$$

The remaining states from this intersection comprise the available system states and can be represented as follows:

$$(0 \ \Delta \ \Delta)$$
$$(2 \ \Delta \ \Delta).$$

The second system submode is now logically intersected with these available system states. This can be shown as

$$(\Delta \ 1 \ 1) \cap (0 \ \Delta \ \Delta) = (0 \ 1 \ 1)$$
$$(\Delta \ 1 \ 1) \cap (2 \ \Delta \ \Delta) = (2 \ 1 \ 1).$$

This process of taking the logical intersection of successive system submodes with the current available system states is continued until all the system submodes have been considered. In the case of the example, the process is complete.

The system states that make up the example system open modes are represented by the results from the above intersections;

$$(1 \quad \Delta \quad \Delta)_1$$

$$(0 \quad 1 \quad 1)_2$$

$$(2 \quad 1 \quad 1)_2 \,.$$

Note that these are mutually exclusive. The subscripts on the rows are used here to indicate which system submode yielded that particular row. The information required for the generation of the System Mode Model is now completed.

2.3 SYSTEM MODE MODEL

The System Mode Model is an algebraic equation for the probability of occurence of the system mode under consideration. This is obtained by summing the probabilities of occurrence for the mutually exclusive rows generated by the STRIP process.

The System Mode Model for the example system is given as follows:

$$P(\text{SYSTEM OPEN MODE}) = P(1,1) + P(1,0) x P(2,1) x P(3,1) + P(1,2) x P(2,1) x P(3,1) .$$

By factoring $P(2,1)xP(3,1)$ from the second and third terms, the equation can be rewritten as

$$P(\text{SYSTEM OPEN MODE}) = P(1,1) + [P(1,0) + P(1,2)] \text{ x } P(2,1) x P(3,1) .$$

The relationship of terms in the System Mode Model to submodes in the System Mode Array can be made in the following manner. The Basic Submode Probability will be defined as the probability of occurrence of a system submode, regardless of any overlapping. The Basic Submode Probabilities for the example are given as

$BP_1 = P(1,1)$ for the first system submode

and

$$BP_2 = P(2,1) \times P(3,1) \text{ for the second system submode.}$$

The probability of occurrence of the system submodes resulting from the STRIP process, which has removed any overlapping of the original SMA submodes, is called the Total Row Probability. For the first system submode, the Total Row Probability is given by

$$TR_1 = P(1,1) = BP_1 \ .$$

The Total Row Probability for the second system submode is given by

$$\begin{aligned} TR_2 &= \left[P(1,0)+P(1,2)\right] x P(2,1) x P(3,1) \\ &= \left[P(1,0)+P(1,2)\right] x BP_2 \ . \end{aligned}$$

The System Mode Model can now be rewritten as a series of algebraic equations which collectively are identical to the above model for P(SYSTEM OPEN MODE). For the example, the System Mode Model is written as follows:

$$\begin{aligned} BP_1 &= P(1,1) \\ BP_2 &= P(2,1) x P(3,1) \\ TR_1 &= BP_1 \\ TR_2 &= \left[P(1,0)+P(1,2)\right] x BP_2 \\ P(\text{SYSTEM OPEN MODE}) &= TR_1 + TR_2 . \end{aligned}$$

This form of the System Mode Model is used because it lends itself to computerized implementation. Note that since the Total Row Probabilities are mutually exclusive (the result of STRIPing the SMA), there need be no term of the form

$$-TR_1 \, x TR_2$$

in the equation for P(SYSTEM OPEN MODE), as would be required for non-mutually exclusive probabilities.

2.4 RAPID ANALYSIS PHASE

This section describes the phase of the RAPID technique in which the generated probabilistic model for the system reliability is used to calculate system reliability as a function of time. The model for the three-resistor example system is repeated here for easy reference.

$$BP_1 = P(1,1)$$

$$BP_2 = P(2,1)xP(3,1)$$

$$TR_1 = BP_1$$

$$TR_2 = [P(1,0)+P(1,2)]xBP_2$$

$$P(\text{SYSTEM OPEN MODE}) = TR_1 + TR_2.$$

The first step required in order to make use of the model is to determine the element mode probabilities found in the model. For the example system, the following are required

P(1,0); The probability of element 1 being in the 0 (good) mode,
P(1,1); The probability of element 1 being in the 1st (open) mode,
P(1,2); The probability of element 1 being in the 2nd (shorted) mode,
P(2,1); The probability of element 2 being in the 1st (open) mode, and
P(3,1); The probability of element 3 being in the 1st (open) mode.

Most element mode probabilities are dependent on the length of time the element has been operating, from the time the element was placed into operation ($t = 0$) until some later time, t. The distributions of element mode probabilities as a function of time are therefore required.

2.4.1 Probability Distributions

The cumulative probability of occurrence of event as a function of time will be referred to as the probability distribution for that event. The mathematical expression describing this distribution for the modes of the various system elements is required in the analysis phase of RAPID. A great deal of investigation has been carried out in the field of reliability to determine these functions for various component parts and equipments.

It has been generally shown that most electronic components follow an exponential distribution for the probability of failure at time t, having begun operation at $t = 0$. That is,

$$Q(t) = 1-e^{-\lambda t}, \; \lambda = \text{constant failure rate}.$$

Other types of components may follow different distributions; however, Figure B-2 summarizes the probability distributions for the occurrence of a failure, that have been found to be applicable for various types of components.

From the nature of failure distributions, it can be seen that the model for a system must be evaluated for each point in time that the reliability value is desired. This is required, since the element mode probabilities vary with time.

The three resistor example system model will be evaluated to illustrate the steps in RAPID analysis. Assume that the resistors follow the same probability distribution of failure given by

$$Q(t) = 1-e^{-\lambda Ktx10^{-6}} .$$

In addition, let

$$\lambda = 10 \text{ failures per } 10^6 \text{ hrs. (generic failure rate)}$$

and

$$K = 10 \text{ (environmental factor)},$$

and assume that the open and short modes of the resistors occur in the following proportions:

> Of all possible resistor failures occurring in a given time interval, 90% will be in the open mode and 10% will be in the shorted mode. These modes are mutually exclusive and independent.

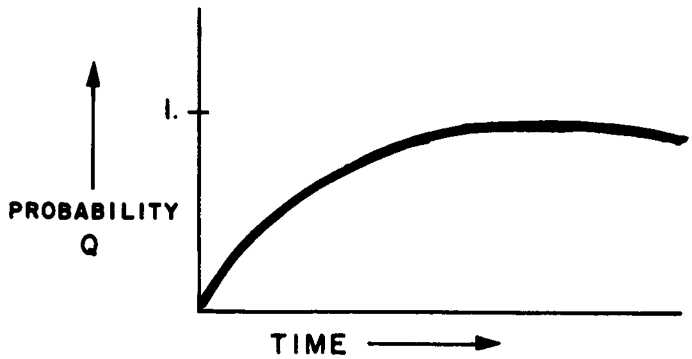
The probability distribution of the open and shorted modes can now be written as

$$Q_o (t) = 1-e^{-.9 \lambda Ktx10^{-6}} = 1-e^{-90tx10^{-6}}$$

$$Q_s (t) = 1-e^{-.1 \lambda Ktx10^{-6}} = 1-e^{-10tx10^{-6}} .$$

EXPONENTIAL

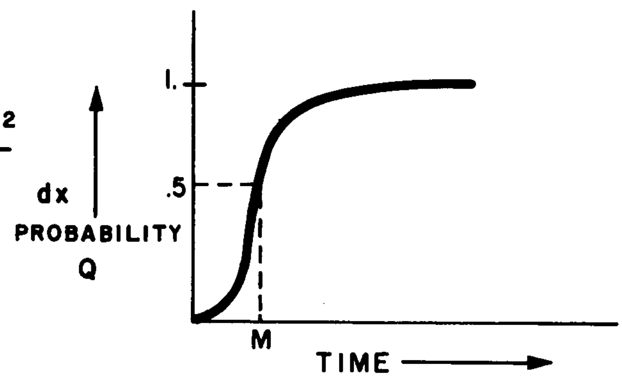
$$Q(t) = 1 - e^{-\lambda t}$$



NORMAL

$$Q(t) = \frac{1}{\alpha\sqrt{2\pi}} \int_0^t e^{-\frac{(X-M)^2}{2\sigma^2}} dx$$

σ = standard deviation



WEIBULL (2 parameter)

$$Q(t) = 1 - e^{-\left(\frac{t}{\alpha}\right)^{\beta}}$$

α = scaling parameter; $\alpha > 0$

β = shaping parameter; $\beta > 0$

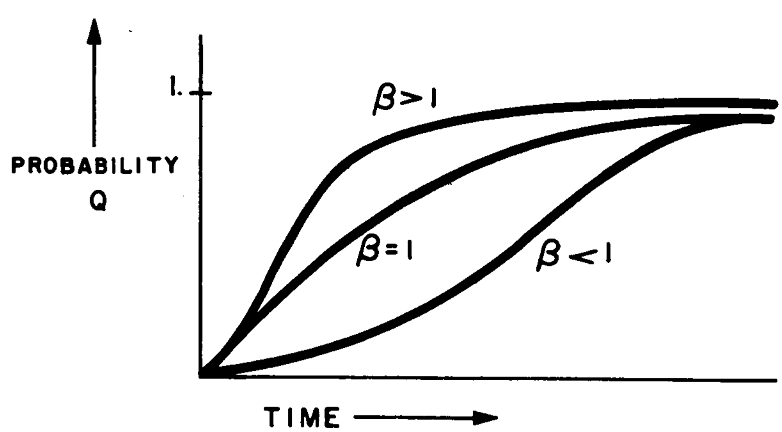


Figure B-2. Sample Cumulative Probability Functions

Also, the probability of the "good" resistor mode is obviously

$$1-Q(t) = e^{-\lambda Ktx10^{-6}} = e^{-100tx10^{-6}}$$

since

P(GOOD) + P(OPEN OR SHORT) = 1.

The model for the example system will be evaluated at time (t=10), given that the system was operating satisfactorily at t=0. The element mode probabilities have been found to be

$$P(1,0) = 1-Q(t) = e^{-100tx10^{-6}} = e^{-1000x10^{-6}} = .999$$

$$P(1,1) = P(2,1) = P(3,1) = Q_o(t) = 1-e^{-90tx10^{-6}} = 1-e^{-900x10^{-6}} = .0009$$

$$P(1,2) = Q_s(t) = 1-e^{-10tx10^{-6}} = 1-e^{100x10^{-6}} = .0001.$$

Note that

$$\sum_{k_i=0}^{K} P(1,k) = 1,$$

as pointed out in Section 2.1.1.1 above. The mode probabilities are used in the model as follows

$$BP_1 = P(1,1) = .0009$$

$$BP_2 = P(2,1)xP(3,1) = (.0009)^2 = .81x10^{-6}$$

$$TR_1 = BP_1 = .0009$$

$$TR_2 = \left[P(1,0)+P(1,2)\right] xBP_2 = (.9991)x.81x10^{-6} \cong .81x10^{-6}$$

$$P(\text{SYSTEM OPEN MODE @ } t = 10) = TR_1 + TR_2 = .0009 + .81x10^{-6} \cong .000901.$$

3 COMPUTER IMPLEMENTATION

The RAPID technique as described in this report has been fully implemented on two computers. The two Reliability Simulators differ in the manner of implementing the RAPID technique because of differences in the two computer systems. The two simulators are presented in detail in the following sections.

Section 3.1 describes the contents and use of Data Transmittal Forms which have been designed for inputting information to the computer programs. Sections 3.2 and 3.3 describe the methodologies and programs used to implement the RAPID technique on the IBM 7090 and the IBM 1620 systems, respectively.

3.1 DATA TRANSMITTAL

Explicit instructions are presented in this section for recording reliability data in order to obtain a RAPID analysis. Forms have been developed from which the data are directly recorded for computer utilization. These Data Transmittal Forms (DTF) are identified by a RAPID DTF number located in the title block of each form.

The blocks on each form are representative of the columns of an eighty-column tabulation card. Each row on the DTF forms represents a card. There are four terminologies used in this report with which the reader may not be familiar. These are:

a. Floating point number,
b. Left justification,
c. Right justification, and
d. Alphanumeric characters.

A floating point number may be defined as one with a decimal point at the beginning, at the end, or between any two digits of the number. Left justification is the positioning of the left-most alphanumeric character in the left-most position of the available field. Similarly, right justification is the positioning the right-most alphanumeric character in the right-most position of the available field. Alphanumeric characters consist of all the alphabetic characters (A through Z), all numerical characters (0 through 9) and ten special characters: (.), (,), (+), ($), (*), (-), (/), ()), ((), (=).

3.1.1 Data Transmittal Form I

Data Transmittal Form I defines the system under consideration. The system definition includes the following:

- TITLE CARD I

CODE:	The CODE is a reference name under which probabilistic results are permanently stored.
SYSTEM NAME:	The SYSTEM NAME is the name assigned to the system under consideration.
DATE:	The DATE represents the time at which the analysis was performed.

- TITLE CARD II

H:	The H is a System Mode Symbol which represents the particular system mode.
MODE NAME:	The MODE NAME is the name of the system mode under consideration.

- SYSTEM DEFINITION CARD

NE:	The NE represents the number of elements in the system.
SYSTEM DEFINITION ARRAY:	The SYSTEM DEFINITION ARRAY defines all the possible system states. The entries of this array are the number of specified element failure modes for each element in the system.

- TIME CARDS

NT:	The NT represents the number of transition times in the system mission.
TIME:	The TIME represents the particular transition times. These are normally the times at which the environmental factors change.
J:	The J represents the number of desired equal intermediate times at which output data are requested in the interval defined by the bounding transition times.

DATA TRANSMITTAL FORM I

TITLE CARD 1

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
4 thru 11	CODE	Alpha Numeric	No Restrictions	System Name Reference Code
18 thru 63	SYSTEM NAME	Alpha Numeric	No Restrictions	System Name
70 and 71 73 and 74 76 and 77	DATE	Integer	Right Justified	Date: Month Day Year
80	None	1	Pre-Assigned	Control Number

TITLE CARD 2

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
3	H	Integer	No Restrictions	System Mode Symbol
18 thru 63	MODE NAME	Alpha Numeric	No Restrictions	System Mode Name
80	None	2	Pre-Assigned	Control Number

SYSTEM DEFINITION CARD

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
1 thru 3	NE	Integer	Right Justified	Number of Elements
21 thru 70	ELEMENT NUMBER	Integer	Left Justified	System Definition Array
80	None	3	Pre-Assigned	Control Number

ALL SHADED COLUMNS MUST BE LEFT BLANK

DATA TRANSMITTAL FORM I (cont)

CARD 1

TIME CARDS

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
1 thru 3	NT	Integer	Right Justified	Number of Transition Times
5 thru 14	TIME	Floating Point Number	Left Justified	Initial Time - T_0 in hours If T_0 is a zero, it must be entered as "0."
16 thru 18	J	Integer	Right Justified	Number of desired equal intermediate times between T_0 and T_1
20 thru 29	TIME	Floating Point Number	Left Justified	Transition Time - T_1 in hours
31 thru 33	J	Integer	Right Justified	Number of desired equal intermediate times between T_1 and T_2
35 thru 44	TIME	Floating Point Number	Left Justified	Transition Time - T_2 in hours
46 thru 48	J	Integer	Right Justified	Number of desired equal intermediate times between T_2 and T_3
50 thru 59	TIME	Floating Point Number	Left Justified	Transition Time - T_3 in hours
61 thru 63	J	Integer	Right Justified	Number of desired equal intermediate times between T_3 and T_4
65 thru 74	TIME	Floating Point Number	Left Justified	Transition Time - T_4 in hours
76 thru 78	J	Integer	Right Justified	Number of desired equal intermediate times between T_4 and T_5
80	None	4	Pre-Assigned	Control Number

ALL SHADED COLUMNS MUST BE LEFT BLANK

DATA TRANSMITTAL FORM I (cont)

CARDS 2, 3, 4, 5

TIME CARDS

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
5 thru 14	TIME	Floating Point Number	Left Justified	Transition Time - T_5; T_{10}; T_{15}; T_{20} in hours
16 thru 18	J	Integer	Right Justified	Number of desired equal intermediate times between T_5 & T_6; T_{10} & T_{11}; T_{15} & T_{16}; T_{20} & T_{21}
20 thru 29	TIME	Floating Point Number	Left Justified	Transition Time - T_6; T_{11}; T_{16}; T_{21} in hours
31 thru 33	J	Integer	Right Justified	Number of desired equal intermediate times between T_6 & T_7; T_{11} & T_{12}; T_{16} & T_{17}; T_{21} & T_{22}
35 thru 44	TIME	Floating Point Number	Left Justified	Transition Time - T_7; T_{12}; T_{17}; T_{22} in hours
46 thru 48	J	Integer	Right Justified	Number of desired equal intermediate times between T_7 & T_8; T_{12} & T_{13}; T_{17} & T_{18}; T_{22} & T_{23}
50 thru 59	TIME	Floating Point Number	Left Justified	Transition Time - T_8; T_{13}; T_{18}; T_{23} in hours
61 thru 63	J	Integer	Right Justified	Number of desired equal intermediate times between T_8 & T_9; T_{13} & T_{14}; T_{18} & T_{19}; T_{23} & T_{24}
65 thru 74	TIME	Floating Point Number	Left Justified	Transition Time - T_9; T_{14}; T_{19}; T_{24} in hours
76 thru 78	J	Integer	Right Justified	Number of desired equal intermediate times between T_9 & T_{10}; T_{14} & T_{15}; T_{19} & T_{20} (Note 1)
80	None	4	Pre-Assigned	Control Number

NOTE 1: The last entry in the above sequence must be a time entry.

ALL SHADED COLUMNS MUST BE LEFT BLANK

LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF I	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET ___ of ___

TITLE CARD 1

CODE — SYSTEM NAME — DATE — 1

TITLE CARD 2

H — MODE NAME — 2

SYSTEM DEFINITION CARD

NE — ELEMENT NUMBER

1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 20 1 2 3 4 5 6 7 8 9 30 1 2 3 4 5 6 7 8 9 40 1 2 3 4 5 6 7 8 9 50

3

TIME CARDS

NT — TIME — J — TIME — J — TIME — J — TIME — J — TIME — J

4
4
4
4
4

3.1.2 Data Transmittal Form II

Data Transmittal Form II defines the System Mode under consideration. The system mode definition includes the following:

- SYSTEM MODE ARRAY CARDS

NS:	The NS represents the number of System Submodes (rows) contained in the System Mode Array.
SYSTEM MODE ARRAY:	The SYSTEM MODE ARRAY (SMA) defines the system mode under consideration. The SMA consists of System Submodes which are an ordered sequence of element modes.

DATA TRANSMITTAL FORM II

SYSTEM MODE ARRAY CARDS

CARD 1

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
1 thru 3	NS	Integer	Right Justified	Number of rows in System Mode Array
21 thru 70	ELEMENT NUMBER	Integer	No Restrictions	System Submode
80	None	5	Pre-Assigned	Control Number

CARDS 2, 3, 4, . . . , NS (Number of Rows in Array)

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
21 thru 70	ELEMENT NUMBER	Integer	No Restrictions	System Submode
80	None	5	Pre-Assigned	Control Number

ALL SHADED COLUMNS MUST BE LEFT BLANK

LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF II	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET ____ of ____

SYSTEM MODE ARRAY CARDS

NS (card columns 1–3)

ELEMENT NUMBER (card columns 21–70): 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 20 1 2 3 4 5 6 7 8 9 30 1 2 3 4 5 6 7 8 9 40 1 2 3 4 5 6 7 8 9 50

Card columns: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

Column 80 (each of 20 rows): 5

3.1.3 Data Transmittal Form III

Data Transmittal Form III defines the elements comprising the system. The information required to define these elements includes the following:

- ELEMENT CARDS

EN:	The EN represents the Element Number which is assigned to a particular element in the analysis.
NAME:	The NAME represents the name of the particular element.
S:	The S conveys the manner in which the element mode probabilities are to be determined.
LAMBDA:	The LAMBDA represents the failure rate per million hours. This entry is required only if the probabilities are to be computed from an exponential failure distribution.
CODE:	The CODE represents a name under which the element mode probabilities are stored.
MODE NAME:	The MODE NAME represents the name assigned to a particular element mode.
C:	The C represents the conditional probability of occurrence of the particular element mode.

DATA TRANSMITTAL FORM III

CARD 1 - BLOCK 1

ELEMENT CARDS

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
2 thru 3	EN	Integer	Right Justified	Element Number
5 thru 14	NAME	Alpha Numeric	No Restrictions	Element Name
16	S	0,1,2	No Restrictions	0 Implies element mode probabilities are stored under code name (ref. columns 30-37). 1 Implies element mode probabilities are computed by an exponential failure rate. 2 Implies element mode probabilities are given as inputs.
18 thru 27	LAMBDA	Floating Point Number	Left Justified	Failure rate per million hours If entry in column 16 is 0, make no entry. If entry in column 16 is 1, enter failure rate. If entry in column 16 is 2, make no entry.
30 thru 37	CODE	Alpha Numeric	No Restrictions	If entry in column 16 is 0, then enter System Name Reference Code (Columns 4-11; Form I; Title Card 1) of appropriate System Analysis Data Forms. If entry in column 16 is 1, then enter Element Name Code. If entry in column 16 is 2, then enter Element Name Code.
80	None	6	Pre-Assigned	Control Number

ALL SHADED COLUMNS MUST BE LEFT BLANK

CARDS 2, 3, 4, - BLOCK 1

ELEMENT CARDS

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
5	None	1,4,7	Pre-Assigned	1 in Card 2 identifies Mode Name 1 4 in Card 3 identifies Mode Name 4 7 in Card 4 identifies Mode Name 7
7 thru 21	MODE NAME	Alpha Numeric	No Restrictions	Mode Name 1 (Card 2) Mode Name 4 (Card 3) Mode Name 7 (Card 4)
23 thru 28	C	Floating Point Number	Left Justified	If entry in column 16 (Card 1) is 0, make no entry. If entry in column 16 (Card 1) is 1, enter conditional probability of failure for the mode in columns 7-21 for Cards 2,3,4. If entry in column 16 (Card 1) is 2, make no entry.
30	None	2,5,8	Pre-Assigned	2 in Card 2 identifies Mode Name 2 5 in Card 3 identifies Mode Name 5 8 in Card 4 identifies Mode Name 8
32 thru 46	MODE NAME	Alpha Numeric	No Restrictions	Mode Name 2 (Card 2) Mode Name 5 (Card 3) Mode Name 8 (Card 4)
48 thru 53	C	Floating Point Number	Left Justified	If entry in column 16 (Card 1) is 0, make no entry. If entry in column 16 (Card 1) is 1, enter conditional probability of failure for the mode in columns 32-46 for Cards 2,3,4. If entry in column 16 (Card 1) is 2, make no entry.
55	None	3,6,9	Pre-Assigned	3 in Card 2 identifies Mode Name 3 6 in Card 3 identifies Mode Name 6 9 in Card 4 identifies Mode Name 9
57 thru 71	MODE NAME	Alpha Numeric	No Restrictions	Mode Name 3 (Card 2) Mode Name 6 (Card 3) Mode Name 9 (Card 4)
73 thru 78	C	Floating Point Number	Left Justified	If entry in column 16 (Card 1) is 0, make no entry. If entry in column 16 (Card 1) is 1, enter conditional probability of failure for the mode in columns 57-71 for Cards 2,3,4. If entry in column 16 (Card 1) is 2, make no entry.
80	None	6	Pre-Assigned	Control Number

NOTE: Use additional blocks for succeeding elements.

ALL SHADED COLUMNS MUST BE LEFT BLANK

LSI	LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF III	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET ___ of ___

ELEMENT CARDS

EN	NAME	S	LAMBDA	CODE	
					6

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

	MODE NAME	C		MODE NAME	C		MODE NAME	C	
1			2			3			6
4			5			6			6
7			8			9			6

EN	NAME	S	LAMBDA	CODE	
					6

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

	MODE NAME	C		MODE NAME	C		MODE NAME	C	
1			2			3			6
4			5			6			6
7			8			9			6

EN	NAME	S	LAMBDA	CODE	
					6

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

	MODE NAME	C		MODE NAME	C		MODE NAME	C	
1			2			3			6
4			5			6			6
7			8			9			6

3.1.4 Data Transmittal Form IIIA

Data Transmittal Form IIIA conveys the information which is essential when the element mode probabilities are computed from an exponential distribution. The information required includes the following:

- ELEMENT K FACTOR CARDS

EB:	The EB represents the element number which is associated with the particular block of K factors.
EE:	Occasionally the K factors of several elements are identical. If this occurs, and if these elements succeed each other, then the first and last element numbers of this sequence are placed in EB and EE respectively.
K FACTOR:	The K FACTOR represents the environmental factor for the corresponding sequential transition time interval.

DATA TRANSMITTAL FORM IIIA

ELEMENT K FACTOR CARDS (NOTE 1)

CARD 1 - BLOCK 1

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
2 thru 3	EB	Integer	Right Justified	Element Number
6 thru 7	EE	Integer	Right Justified	If the elements succeeding EB have identical K Factors, then the last Element Number in this sequence is entered. If the succeeding element is not identical to EB, enter EB entry again in column EE.
10 thru 19	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_0, T_1)$
24 thru 33	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_1, T_2)$
38 thru 47	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_2, T_3)$
52 thru 61	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_3, T_4)$
66 thru 75	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_4, T_5)$
80	None	7	Pre-Assigned	Control Number

CARDS 2, 3, 4, 5 - BLOCK 1

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
10 thru 19	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_5, T_6)$; $[T_{10}, T_{11})$; $[T_{15}, T_{16})$; $[T_{20}, T_{21})$
24 thru 33	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_6, T_7)$; $[T_{11}, T_{12})$; $[T_{16}, T_{17})$; $[T_{21}, T_{22})$
38 thru 47	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_7, T_8)$; $[T_{12}, T_{13})$; $[T_{17}, T_{18})$; $[T_{22}, T_{23})$
52 thru 61	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_8, T_9)$; $[T_{13}, T_{14})$; $[T_{18}, T_{19})$; $[T_{23}, T_{24})$
66 thru 75	K FACTOR	Floating Point Number	Left Justified	K Factor for Time Interval $[T_9, T_{10})$; $[T_{14}, T_{15})$; $[T_{19}, T_{20})$; $[T_{24}, T_{25})$
80	None	7	Pre-Assigned	Control Number

NOTE 1: This form is to be completed only if entry in column 16, Card 1, Form III, is 1, and additional blocks are used for additional elements.

ALL SHADED COLUMNS MUST BE LEFT BLANK

LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF IIIA	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET ___of___

ELEMENT K FACTOR CARDS

EB (2–3)	EE (6–7)	K FACTOR (10–19)	K FACTOR (24–33)	K FACTOR (38–47)	K FACTOR (52–61)	K FACTOR (66–75)	80
							7
							7
							7
							7
							7

EB (2–3)	EE (6–7)	K FACTOR (10–19)	K FACTOR (24–33)	K FACTOR (38–47)	K FACTOR (52–61)	K FACTOR (66–75)	80
							7
							7
							7
							7
							7

EB (2–3)	EE (6–7)	K FACTOR (10–19)	K FACTOR (24–33)	K FACTOR (38–47)	K FACTOR (52–61)	K FACTOR (66–75)	80
							7
							7
							7
							7
							7

3.1.5 Data Transmittal Form IIIB

Data Transmittal Form IIIB conveys the element mode probabilities which are discrete inputs. These probabilities are distinguished from the computed ones in that they may not be related to any known distribution. These probabilities must be given for all the transition times and the intermediate times. The information required includes the following:

- ELEMENT MODE PROBABILITY CARDS

EN:	The EN represents the particular element number.
M:	The M represents the particular element mode.
TIME:	The TIME represents the particular time.
PROBABILITY:	The PROBABILITY represents the probability of occurrence of the particular element mode at the corresponding time.

3.1.6 Data Transmittal Form IV

Data Transmittal Form IV serves as an aid to the analyst. This form is not required, but it is often useful for discussion and reference.

DATA TRANSMITTAL FORM IIIB

ELEMENT MODE PROBABILITY CARD (NOTE 1)

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
2 thru 3	EN	Integer	Right Justified	Element Number
6	M	Integer	No Restrictions	Element Mode
9 thru 18	TIME	Floating Point Number	Left Justified	Time (hours)
20 thru 29	PROBABILITY	Floating Point Number	Left Justified	Element Mode Probability
33 thru 42	TIME	Floating Point Number	Left Justified	Time (hours)
44 thru 53	PROBABILITY	Floating Point Number	Left Justified	Element Mode Probability
57 thru 66	TIME	Floating Point Number	Left Justified	Time (hours)
68 thru 77	PROBABILITY	Floating Point Number	Left Justified	Element Mode Probability
80	None	8	Pre-Assigned	Control Number

NOTE 1: This form must be filled out only for the elements which have a 2 in the "S" column (16) of Form III, Card 1.

ALL SHADED COLUMNS MUST BE LEFT BLANK

LSI	LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF IIIB	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET ____ of ____

ELEMENT MODE PROBABILITY CARDS

	EN		M		TIME		PROBABILITY		TIME		PROBABILITY		TIME		PROBABILITY		
1	2 3	4 5	6	7 8	9 10 11 12 13 14 15 16 17 18	19	20 21 22 23 24 25 26 27 28 29	30 31 32	33 34 35 36 37 38 39 40 41 42	43	44 45 46 47 48 49 50 51 52 53	54 55 56	57 58 59 60 61 62 63 64 65 66	67	68 69 70 71 72 73 74 75 76 77	78 79	80
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8
																	8

PROBABILISTIC SYSTEM MODE ANALYSIS

SHEET ___ of ___

RAPID DTF IV

SYSTEM DIAGRAM

3.2 IBM 7090 VERSION

3.2.1 Introduction

This section describes the methodology for system reliability simulation as written for the IBM 7090 computer. Each step is illustrated relative to a hypothetical system. A main computer program controls the simulation, calling for subroutines to perform the required operations as each data set is processed. All references to computer source statements are in the MAD* language.

Simulation of system reliability is accomplished on the IBM 7090 computer by repeated application of the RAPID technique to automatically generate the desired probabilistic models until all the data describing the system in detail have been exhausted. The models are contained in subroutines which are automatically constructed during the simulation. The execution sequence for the subroutines is "nested" in a manner unique to the sequence in which the input data are processed.

3.2.2 System Description

Figure B-3 is given as an illustration for this discussion. Since the analysis is automatic and continues until all input data have been processed, the system complexity has no effect on the simulation process except to extend the total running time on the computer.

The system is composed of four functional levels. At the subsystem level are three subsystems; SUBS1, SUBS2, SUBS3, the second two of which have additional components at lower levels. These subsystems are the "elements" in the initial data set describing the SYSTEM. These data define a system of 3 elements that is to be simulated. This is designated System Number "1," and is completely simulated by generating a computer subroutine named "SYSTEM" before any lower functional levels are considered.

The subroutine generated by the simulator uses the probabilities for the various modes of the three elements SUBS1, SUBS2, and SUBS3. These probabilities are defined in the SYSTEM subroutine by statements calling either for computations using the exponential function or for execution of other subroutines which are generated later in the simulation procedure. It should be noted that these generated subroutines are not being executed during the simulation. They are being constructed -- statement by statement -- for use in a reliability calculation at a later time when simulation is completed.

---

*Michigan Algorithm Decoder; University of Michigan Executive System for the IBM 7090 Computer, Ann Arbor, Michigan

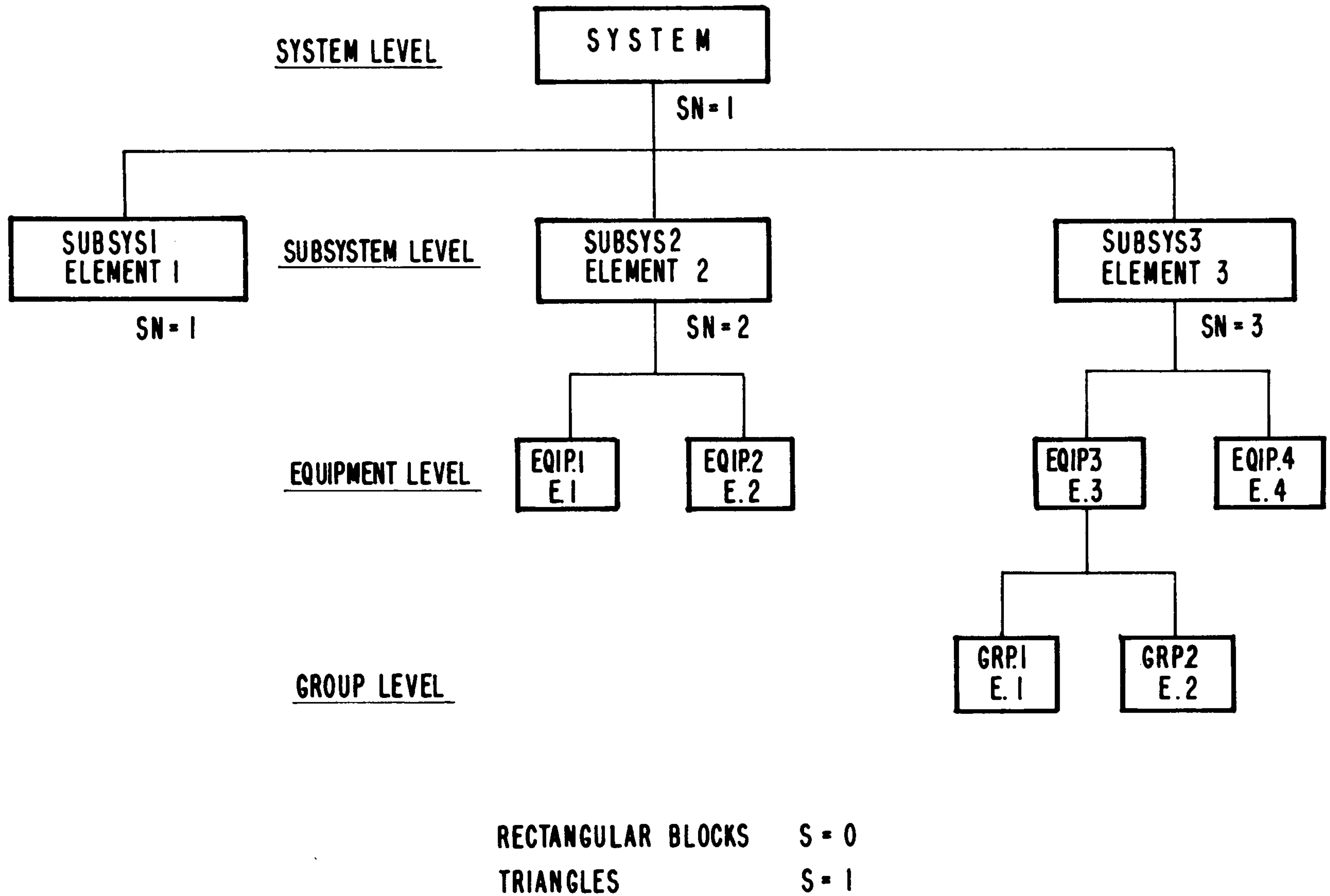


Figure B-3. Example System for IBM 7090 Reliability Simulator Discussion

By noting the S codes (DTF III, col. 16) for the SYSTEM components in Figure B-3, it can be seen that four separate simulations are performed. That is, four subroutines having the names SYSTEM, SUBS2, SUBS3, and EQUIP3 are generated by the simulator. These are nested in their calling sequence so that a single execution statement is sufficient to compute the system reliability when the simulation is completed. This statement is

"EXECUTE SYSTEM. (PFAIL)".

The returned argument, PFAIL, is the probability of the SYSTEM failure mode occurring for the mission described by the TIME CARDS of the SYSTEM input data.

3. 2. 3 System Inputs

The RAPID Analysis Data Transmittal Forms are used for inputting the data for simulation. However, there are two differences in the entry of certain data on the forms:

a. DATA TRANSMITTAL FORM I (FIGURE B-4)
TITLE CARD 1
SN: The SN is the System Number of the system under consideration. (Allowed characters: Integer)

"SN" is an indicator for directing the reliability simulator through the input data. The data are processed in increasing SN order.

b. DATA TRANSMITTAL FORM III (FIGURE B-6)
ELEMENT CARDS
S: The S conveys the manner in which the element is to be modeled. (Allowed characters: 0, 1)

0 Element Probabilistic Model

An S code of "0" implies the element is comprised of components at lower functional levels in the system. No failure rate exists

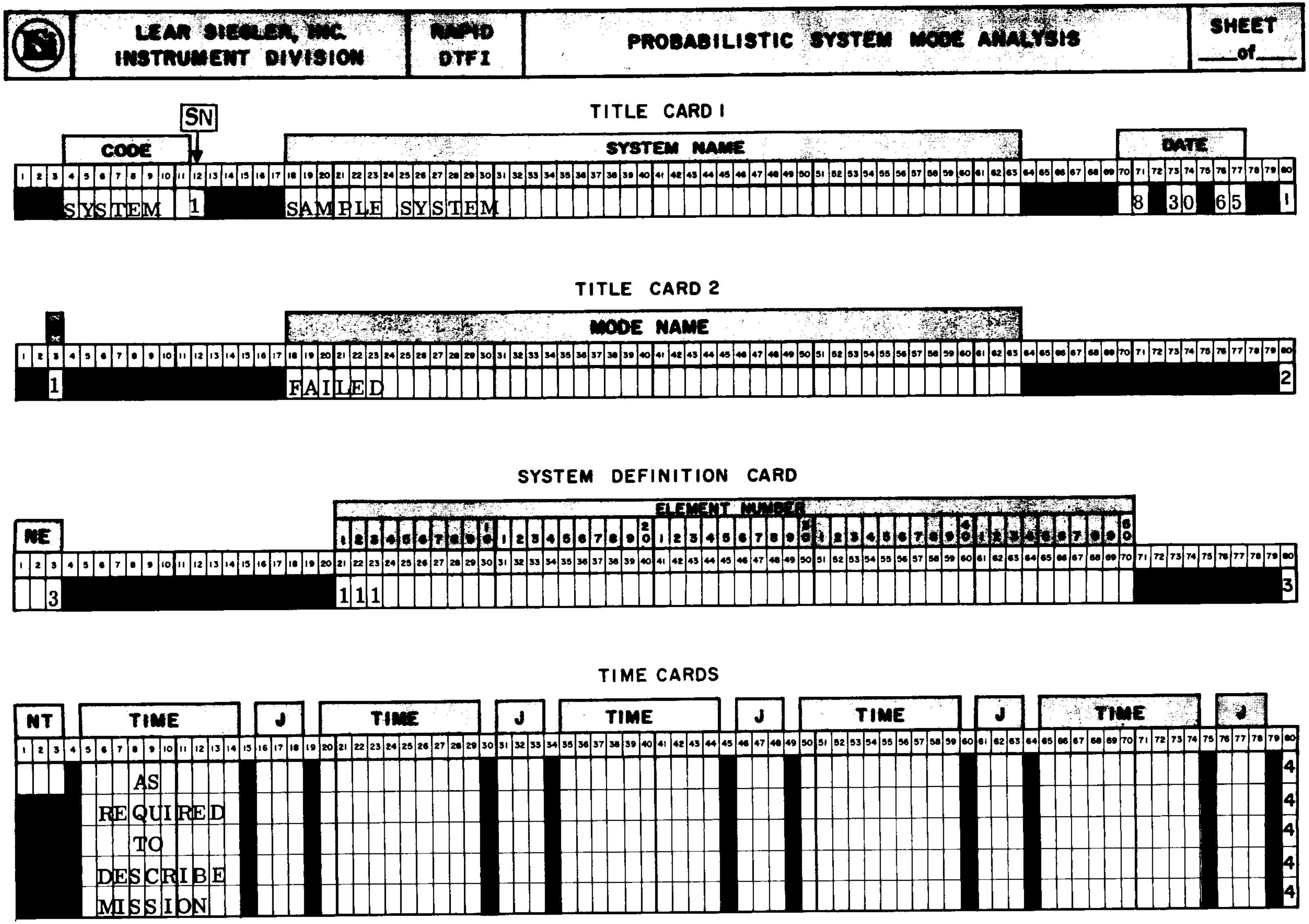


Figure B-4. RAPID Analysis, Data Transmittal Form I (DTF-I) for the element. A subroutine having the name of the element must be generated. The element mode probabilities are computed by this subroutine.

1 Element Exponential Failure Rate Model.

An S code of "1" implies the element has no components at lower functional levels in the system. A failure rate exists for the element. The element mode probabilities are computed using the exponential function.

The input data from DTF I for System Number "1" are given in Figure B-4. For this discussion, all elements will be assumed to have a "good" and a "failed" mode only. Thus, the SDA is given with "1" for each element.

The reliability simulator is capable of handling up to seventeen failure modes for each of the elements having an S code of "1". For the sake of simplicity, however, this discussion is limited to a single failure mode for each element.

The TIME CARDS from DTF 1 follow the SDA in the input deck. These are stored for later use in any calculations of the system reliability. It is required that all TIME CARDS be identical for the various numbered "systems" in the simulator input data.

The SMA and SYSTEM element data are shown in Figures B-5 and B-6.

3.2.4 System Simulation

The input data for System Number "1" are processed by the simulator. The complete data set is read before the RAPID technique is applied to develop the model. The data are related in storage to System Number by the use of double subscripts; the first is the System Number and the second is the element number.

The SYSTEM model is developed and stored on magnetic tape in card image form ready for insertion at a later time in the analysis into the body of the subroutine called "SYSTEM" which is about to be generated.

After the model is stored, the opening statements to the SYSTEM subroutine are written onto another tape. The element data in storage are then analyzed, noting by the S code whether the element is comprised of components at lower functional levels (S=0), or is represented by an element failure rate (S=1).

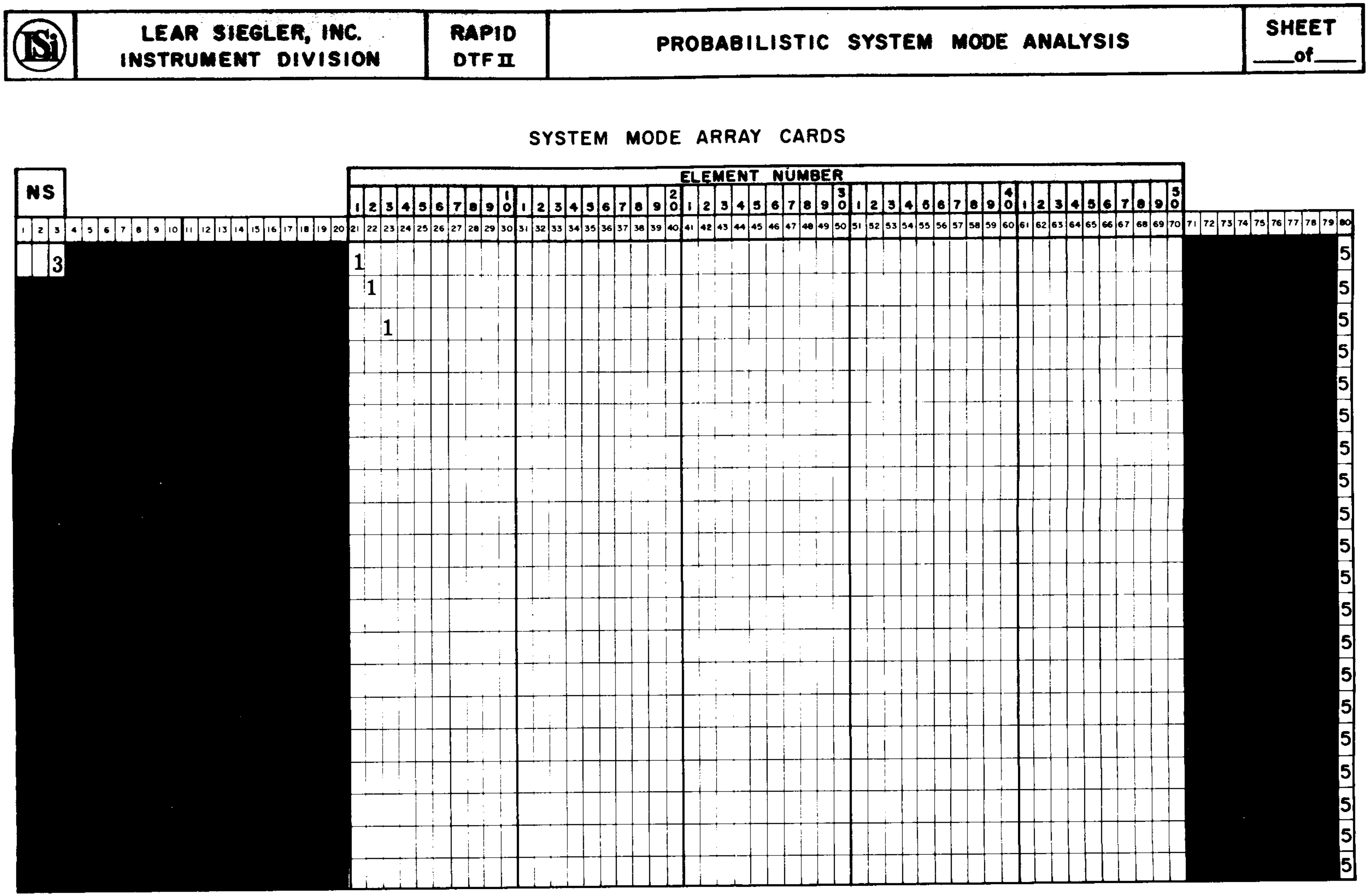


Figure B-5. RAPID Analysis, Data Transmittal Form II (DTF-II)

LSI	LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF III	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET ___ of ___

ELEMENT CARDS

EN	NAME	S	LAMBDA	CODE	80
1	SUBS1	1	XXXX.XX	EL1	6

	MODE NAME	C		MODE NAME	C		MODE NAME	C	80
1	FAILED	1.	2			3			6
4			5			6			6
7			8			9			6

EN	NAME	S	LAMBDA	CODE	80
2	SUBS2	0		EL2	6

	MODE NAME	C		MODE NAME	C		MODE NAME	C	80
1	FAILED	1.	2			3			6
4			5			6			6
7			8			9			6

EN	NAME	S	LAMBDA	CODE	80
3	SUBS3	0		EL3	6

	MODE NAME	C		MODE NAME	C		MODE NAME	C	80
1	FAILED	1.	2			3			6
4			5			6			6
7			8			9			6

Figure B-6. RAPID Analysis, Data Transmittal Form III (DTF-III)

When an S code of "1" is recognized for SUBS1, a statement is inserted into the SYSTEM subroutine calling for another subroutine, which will use the exponential function to compute P(1, 1) and 1-P(1, 1). These are the probability of failure and probability of success, respectively, for element "1" of the SYSTEM.

The data for element "2" (SUBS2) are processed next. Here the S code is "0", and no failure rate is given. This is an indication for the simulator to insert the following statements into the SYSTEM subroutine being constructed:

1) "EXECUTE SUBS2. (PFAIL)
2) P(2, 1) = PFAIL
3) PP(2) = 1. -PFAIL :

The first statement here calls for another subroutine, SUBS2, which will need to be generated later in the analysis. This is how the execution of the various subroutines is "nested" in the SYSTEM subroutine in a sequence unique to the input data.

As each element is reviewed, statements are inserted into the subroutine being developed. The probability of failure and the probability of success for element "2" are defined in the SYSTEM subroutine by the presence of program statements 2) and 3) above.

An identical situation to that described for SUBS2 occurs for element "3" (SUBS3). The S code of "0" results in generation of the following statements for the SYSTEM subroutine:

"EXECUTE SUBS3. (PFAIL)
P(3, 1) = PFAIL
PP(3) = 1. -PFAIL".

Analysis of the input data for describing SUBS1, SUBS2, and SUBS3 as elements of the SYSTEM is now complete. The SYSTEM model is read from the tape where it had been previously stored and is written onto the tape where the SYSTEM subroutine is being developed.

The model resulting from application of the RAPID technique to a particular SDA and SMA has a standard form. That is, the model is given in executable subroutine statements which occur in a standard sequence.

The first series of statements define the probability of occurrence of each row in the SMA, regardless of whether or not "overlap" exists among rows (whether or not the row probabilities are mutually exclusive). These are the basic row probabilities, BP(i).

The next statements in the model are the probabilities of each row occurring with any overlap among rows removed. This reduction of the SMA to contain only rows whose probabilities are mutually exclusive is accomplished by the RAPID logic. These probabilities are known as the total row probabilities, TP(i).

The next series of statements written on the tape where the SYSTEM subroutine is being developed are for summing the total row probabilities to compute the SYSTEM unreliability. In addition, the necessary closing subroutine statements are written at this time.

A complete subroutine called SYSTEM has now been generated and exists on a magnetic tape ready to be translated from the source language to object code. The control program accomplishes this translation and the object version is again stored on a tape until it is needed in a calculation of the SYSTEM reliability. Figure B-7 is the complete SYSTEM subroutine in the source language.

When the analysis at the system level is completed, the next input data to be processed will be for the subsystem functional level of the SYSTEM. Each subsystem which appeared with an S code of "0" when analyzed as a SYSTEM element must now be treated as a "system" itself in order to generate the additional subroutines needed. Thus, the procedure followed in generating the SYSTEM subroutine is applied again.

The initial subsystem data card from DTF I will define the SUBS2 as System Number "2" to be modeled. The SDA and SMA for this system will be as follows:

SDA

$$\begin{bmatrix} 1_1 & 1_2 \end{bmatrix}$$

SMA

$$\begin{bmatrix} 1_1 & \Delta_2 \\ \Delta_1 & 1_2 \end{bmatrix}$$

```
      EXTERNAL FUNCTION(SUMTP)
      ENTRY TO SYSTEM.
      PROGRAM COMMON NEL(10),NS(10),NT,TT(10),T(10*10*15),XIPT(150)
     1C1(10*15),SDA(10*15)
      DIMENSION BP(100),TP(100),P(30*5),PP(30)
      INTEGER J,NEL,NS,NT,SDA
      EXECUTE PEXP.(P,PP,1)
      EXECUTE SUBS2.(PFAIL)
      P(2,1)=PFAIL
      PP(2)=1.-PFAIL
      EXECUTE SUBS3.(PFAIL)
      P(3,1)=PFAIL
      PP(3)=1.-PFAIL
      BP(1)=P(1,1)
      BP(2)=P(2,1)
      BP(3)=P(3,1)
      TP(1)=BP(1)
      TP(2)=BP(2)*(PP(1))
      TP(3)=BP(3)*(PP(1)*PP(2))
      SUMTP=0.
      THROUGH LA,FOR J=1,1,J.G.NS(1)
LA    SUMTP=SUMTP+TP(J)
      FUNCTION RETURN
      END OF FUNCTION
```

NOTES:

1 Opening statements of SYSTEM subroutine

2 Result of finding code = 1 for SUBS1. The three arguments indicate that P(1,1) and PP(1) for system number "1" will be calculated from the exponential function. Note that PP(1) = 1.-P(1,1).

3 Result of finding code = 0 for SUBS2

4 Result of finding code = 0 for SUBS3

5 SYSTEM Model

6 Sum individual row probabilities of SMA

7 Closing subroutine statements

Figure B-7. SYSTEM Subroutine

The SUBS2 model is developed by application of the RAPID technique to these data and it is stored on the temporary tape, just as in the SYSTEM analysis. It can be seen from Figure B-3 that processing the SUBS2 data will yield a subroutine statement calling for the exponential function to compute the element failure and success probabilities for EQUIP1 and EQUIP2. The SUBS2 model is read from the temporary tape and written on the subroutine tape. The summing statements and closing statements are then written to complete the SUBS2 subroutine. See Figure B-8 for the SUBS2 subroutine.

The SUBS3 data result in a call for the exponential function and for a subroutine called EQUIP3. The EQUIP3 subroutine requires the exponential function again.

Figures B-9 and B-10 illustrate the SUBS3 and EQUIP3 subroutines. Review of Figures B-9 through B-12 will make it clear how the execution of all the subroutines is accomplished by a single call, "EXECUTE SYSTEM. (PFAIL)".

3.2.5 Program *

The Reliability Simulator on the IBM 7090 computer is composed of two parts. In order to make the simulator capable of analyzing large systems comprised of elements having numerous modes, some relatively large areas in memory must be reserved, thus requiring the simulator to exist in two core loads. The main program for each of the loads and their associated subprograms are briefly discussed below. The execution sequence of the two core loads is followed for each "system" simulation, with the resulting reliability model being stored on magnetic tape for future use. Following is a listing of the simulator programs with a brief functional description for each one.

Core 1 Main Program:

Figures B-11 and B-12 present the logic-flow chart and program listing for the Core 1 main program. This calls upon the various subprograms indicated to determine the validity of the input information and to prepare it for the second core modeling operations.

Core 1 Subprograms:

RAP01 (Figures B-13 and B-14)

This subprogram reads the cards containing the DTF I information. Checks are made on each input card to assure the data are in the proper order and some symbolic coding of the System Definition Array is performed also.

*All figures mentioned in this section will be found in the end of this section.

```
       EXTERNAL FUNCTION(SUMTP)
       ENTRY TO SUBS2.
       PROGRAM COMMON NEL(10),NS(10),NT,TT(10),T(10*10*15),XIPT(150)
      1C1(10*15),SDA(10*15)
       DIMENSION BP(100),TP(100),P(30*5),PP(30)
       INTEGER J,NEL,NS,NT,SDA
       EXECUTE PEXP.(P,PP,2)
       BP(1)=P(1,1)
       BP(2)=P(2,1)
       TP(1)=BP(1)
       TP(2)=BP(2)*(PP(1))
       SUMTP=0.
       THROUGH LA,FOR J=1,1,J.G.NS(2)
LA     SUMTP=SUMTP+TP(J)
       FUNCTION RETURN
       END OF FUNCTION
```

Figure B-8. Subsystem 2 (SUBS2) Subroutine -- Sample System

```
       EXTERNAL FUNCTION(SUMTP)
       ENTRY TO SUBS3.
       PROGRAM COMMON NEL(10),NS(10),NT,TT(10),T(10*10*15),XIPT(150)
      1C1(10*15),SDA(10*15)
       DIMENSION BP(100),TP(100),P(30*5),PP(30)
       INTEGER J,NEL,NS,NT,SDA
       EXECUTE PEXP.(P,PP,3)
       EXECUTE EQUIP3.(PFAIL)
       P(2,1)=PFAIL
       PP(2)=1.-PFAIL
       BP(1)=P(1,1)
       BP(2)=P(2,1)
       TP(1)=BP(1)
       TP(2)=BP(2)*(PP(1))
       SUMTP=0.
       THROUGH LA,FOR J=1,1,J.G.NS(3)
LA     SUMTP=SUMTP+TP(J)
       FUNCTION RETURN
       END OF FUNCTION
```

Figure B-9. Subsystem 3 (SUBS3) Subroutine -- Sample System

```
       EXTERNAL FUNCTION(SUMTP)
       ENTRY TO EQUIP3.
       PROGRAM COMMON NEL(10),NS(10),NT,TT(10),T(10*10*15),XIPT(150)
      1C1(10*15),SDA(10*15)
       DIMENSION BP(100),TP(100),P(30*5),PP(30)
       INTEGER J,NEL,NS,NT,SDA
       EXECUTE PEXP.(P,PP,4)
       BP(1)=P(1,1)
       BP(2)=P(2,1)
       TP(1)=BP(1)
       TP(2)=BP(2)*(PP(1))
       SUMTP=0.
       THROUGH LA,FOR J=1,1,J.G.NS(4)
LA     SUMTP=SUMTP+TP(J)
       FUNCTION RETURN
       END OF FUNCTION
```

Figure B-10. Equipment 3 (EQUIP3) Subroutine -- Sample System

RAP02 (Figures B-15 and B-16)

This subprogram reads and checks the system Mode Array (DTF II). The data on each SMA card are symbolically coded and stored in three specialized tables. This minimizes the amount of storage taken up by the SMA, since the element supermodes are not stored. Their presence is noted by the sequence of the table entries only.

RAP07 (Figures B-17 and B-18)

This subprogram reads and checks the cards containing the DTF III information; element descriptions. The number of modes for each element as indicated in the SDA, is checked with the inputs. Any inconsistencies will cause a diagnostic comment to be printed and the simulation of the current "system" to be terminated.

The numbering of the RAPID subprograms is not sequential in the Reliability Simulator on the IBM 7090. There are two reasons for this. The first reason is that the number assigned to the subprograms which perform identical functions in the two versions (IBM 1620 and IBM 7090) was kept the same, but the IBM 7090 version does not require some of the IBM 1620 "utility" functions to be performed. The manner in which the function is performed in the subprograms having similar numbers differs considerably between the two versions. The second reason for non-sequential numbering of the subprograms in the IBM 7090 version, is that the operations during the modeling phase are performed in a slightly different order than on the IBM 1620 version.

RAP08 (Figures B-19 and B-20)

This subprogram reads and checks the K factor information (DTF IIIA). Each element having an "S" code equal to "1" on DTF III must have a K factor associated with it for each time interval as defined on the TIME CARDS which were read by RAP01. The inputs must be in sequential order by element number, so that an array can be loaded with the K factors with unique identification of each K with the correct element.

ALOAD (Figures B-21 and B-22)

This subprogram performs the single function of loading an array called "A" with symbolic information relating to the SDA.

Core 2 Main Program: (Figures B-23 and B-24)

After Core 1 has accomplished all the functions described above, the computer memory is loaded with another main program called CORE 2, with its associated subprograms. The only part of CORE 1 that is retained in memory is an area called "Program Common," which contains critical information needed by CORE 2. This serves as a "communication" area between core loads.

The process of constructing the subprogram containing the model for the system being simulated is accomplished in CORE 2. The available information is analyzed to generate the reliability model which is temporarily stored on magnetic tape. The subprogram is then developed, one statement at a time, and the model is inserted at the appropriate point in the process. CORE 2 then calls upon the computer executive system to compile the generated subprogram and store it on magnetic tape.

Core 2 Subprograms:

RAP04 (Figures B-25 and B-26)

This subprogram generates the basic probability equations [BP(i)] for the reliability model, directly from the SMA information. These are the opening n statements of the model, where n is the number of system submodes in the SMA.

RAP03 (Figures B-27 and B-28)

The major logic involved in generating the total probability equations [TP(i)] for the reliability model is incorporated in this subprogram. Each row of the SMA is analyzed to remove the "overlap" among the submodes. This assures that the final system submode probabilities are mutually exclusive in the reliability model. The equation is in symbolic form when RAP03 has completed the submode analysis.

EQLOAD (Figures B-29 and B-30)

This subprogram is called by RAP03 each time a submode has been analyzed to load the submode probability equation onto magnetic tape. This is accomplished by processing the symbolic representation of the equation and placing characters onto the tape in the image of source program statements.

The function of the above programs is only to simulate a system. The execution of the subprograms generated by the simulator in order to evaluate system reliability must be performed in a separate program existing as a third core load. This core is automatically loaded after all models have been generated by CORES 1 and 2.

PEXP (Figures B-31 and B-32)

This subprogram is loaded with CORE 3 to calculate the element mode probabilities from the exponential function, as required. It is actually called by the generated subprograms comprising the system reliability model whenever a probability is required for an element having a failure rate associated with it.

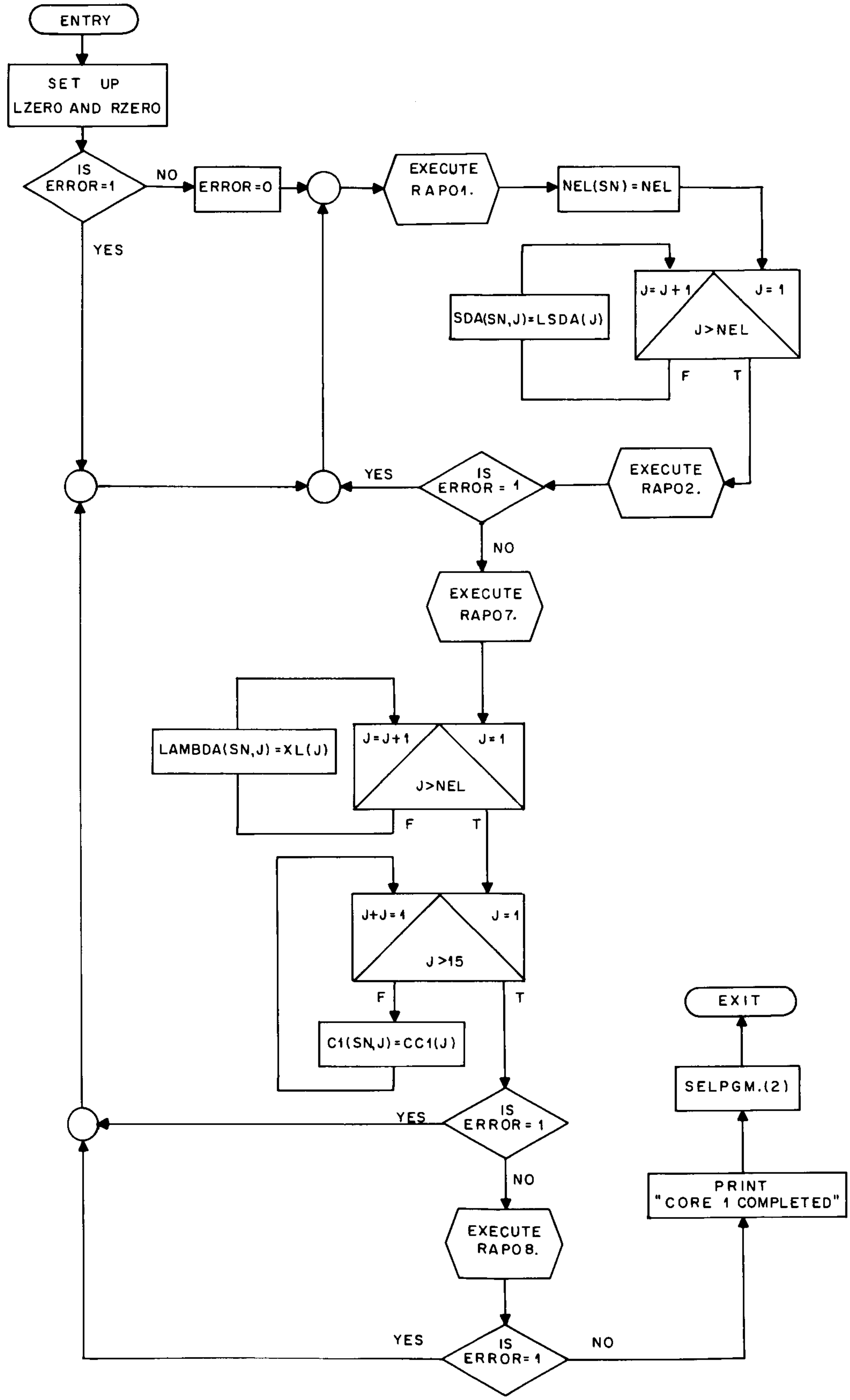


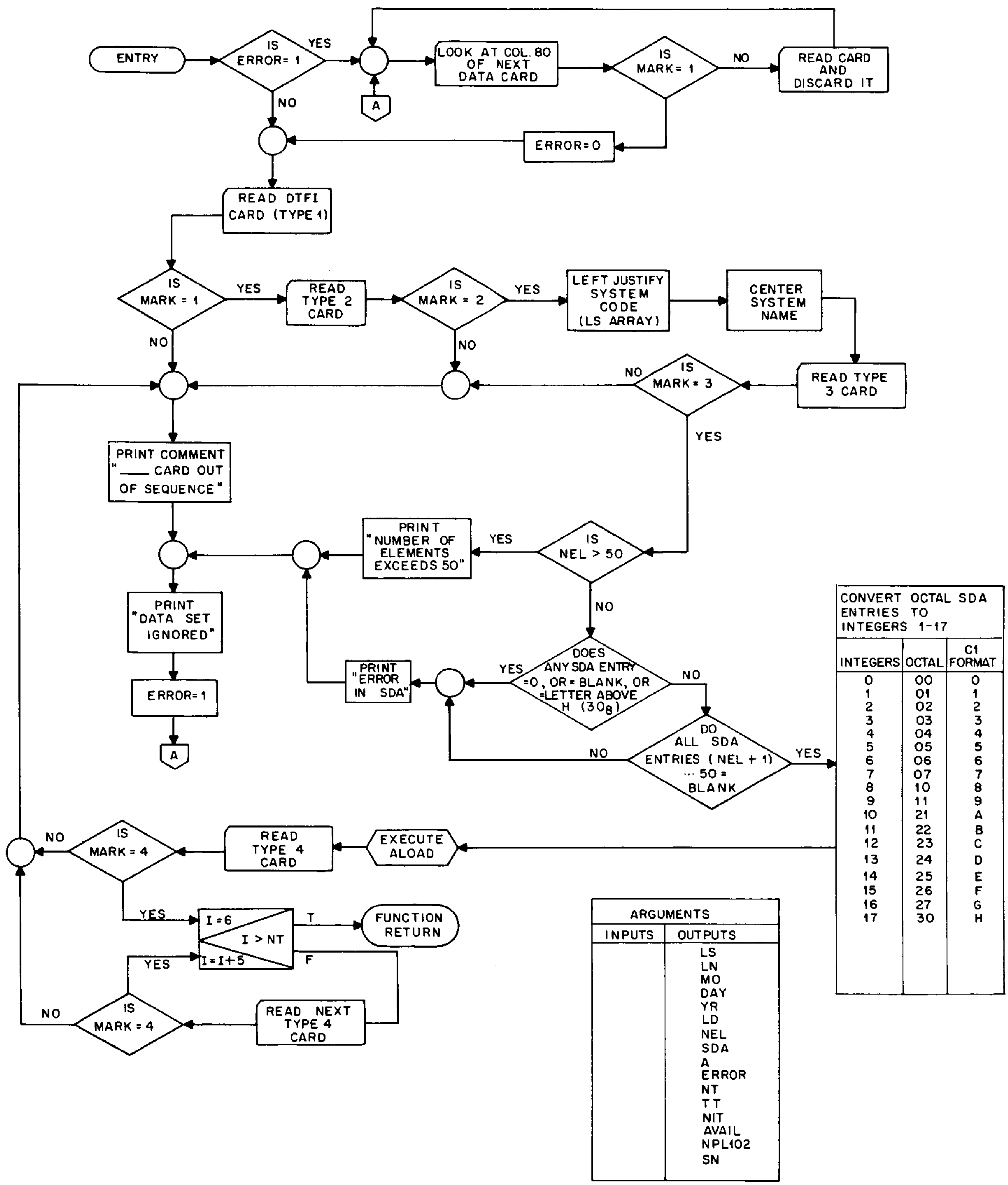
Figure B-11. Core 1 Main Program, Logic Flow Chart

```
R   MAIN PROGRAM FOR RAPID CORE 1
R
R   ....DEFINITION OF PROGRAM VARIABLES....
R
R A(1000,ADIM)            WORKING ARRAY TO STORE RESIDUES DURING STRIP
R ADIM(2)                 DIMENSION VECTOR FOR A.
R AVAIL(25)               ORIGINAL SYMBOLIC SDA FOR CURRENT SYSTEM
R CC1(15)                 CURRENT COND. PROB. LIST
R C1(10*15)               COND. PROB. LIST BY SYSTEM NO.
R CODE(15)                EL. CODE NAMES - CURRENT SYSTEM (6 BCD CHAR.)
R LAMBDA(150,LDIM)        FAILURE RATES BY SYS. NO. AND EL.
R LDIM(2)                 DIMENSION VECTOR FOR LAMBDA - USES SET. SUBR.
R LD(46)                  CURRENT SYSTEM MODE NAME (CENTERED)
R LN(46)                  CURRENT SYSTEM NAME (CENTERED)
R LS(8)                   CURRENT SYSTEM CODE NAME
R LSDA(50)                CURRENT SYSTEM SDA - NOT SYMBOLIC
R LZERO                   SYMBOLIC WORD WITH ZEROES IN LEFT HALF AND 1
R                         BITS IN RT HALF
R MNAME(60*3)             MODE NAMES FOR ELEMENTS - CURRENT SYSTEM
R NAME(30*2)              ELEMENT NAMES - CURRENT SYSTEM
R NEL(30)                 NO. OF ELEMENTS BY SYSTEM NO. (NEL(0)= FOR
R                         CURRENT SYS.)
R NIT(10)                 NO. OF INTERNAL TIMES BETWEEN N AND (N+1) ST
R NT                      NO. OF TRANSITION TIMES
R                         TRANSITION TIMES
R NS(10)                  NO. OF SUBMODES BY SYSTEM NO.
R RZERO                   SYMBOLIC WORD WITH ZEROES IN RT HALF AND 1
R                         BITS IN LEFT HALF
R S(15)                   S CODE FOR ELEMENTS IN CURRENT SYSTEM
R SDA(10*15)              SDA BY SYSTEM AND ELEMENT NO. - NOT SYMBOLIC
R T(10*10*15)             K FACTORS BY SYS. NO., TIME INT., AND EL. NO.
R T1(50)                  ELEMENT DESIGNATION FOR SMA SHORT FORM
R T2(50)                  MODE NO. FOR THE EL. OF IDEN. POSITION IN T1 -
R                         SMA SHORT FORM
R T3(50)                  ORGANIZATION OF SMA BY INC. ORDER OF ROWS -
R                         INDICATES POSITION IN T1 WHERE ROW BEGINS.
R TT(10)                  TRANSITION TIMES
R XL(15)                  CURRENT SYSTEM ELEMENT FAILURE RATES
R
```

Figure B-12. Core 1 Main Program, Program Listing (1 of 2)

```
R    ....PROGRAM LISTING....
R
         N'R
         F'T  TT,LAMBDA,C1,T,CC1,XL
         D'N  CC1(15),NAME(30*2),MNAME(60*3),XL(15)
         V'S  ADIM=2,1,0
         EQUIVALENCE  (ADIM(2),NPL102)
         V'S  LDIM=SET,,0,0
         V'S  ZZZZ= 0,0,0
         P'N  LZERO,RZERO,A(1000,ADIM),ADIM,T1(50),T2(50),T3(50),
        1AVAIL(25),SDA(10*15),NS(10),NEL(30),ERROR,LS(8),LN(46),LD(46)
        2,TT(10),NIT(10),S(15),LAMBDA(150,LDIM),ZZZZ,NT,SN,CODE(15),C1
        3(10*15),T(10*10*15),LSDA(50),MO,DAY,YR
         LZERO=0
         T'H  LOOP,FOR C=0,1,C.G.17
         TEMP=1.LS.C
LOOP     LZERO=LZERO.V.TEMP
         RZERO=LZERO.LS.18
         W'R  ERROR.E.1,T'O RAP1
         ERROR=0
RAP1     EXECUTE  RAP01.(LS,LN,MO,DAY,YR,LD,NEL,LSDA,A,ERROR,NT,TT,NIT,
        1AVAIL,NPL102,SN)
         NEL(SN)=NEL
         T'H  L1,FOR J=1,1,J.G.NEL
L1       SDA(SN,J)=LSDA(J)
         PRINT BCD RESULTS ERROR,LS(1)...LS(8),LN(1)...LN(46),LD(1)...
        1LD(46)
         P'S  NEL,LSDA(1)...LSDA(50),NEL(SN),SDA(SN,1)...SDA(SN,NEL(SN)
        1)
         PRINT OCTAL RESULTS A(1,1)...A(1,(NEL+1)/2)
         EXECUTE ALOAD.(NEL,LSDA,A,AVAIL)
         PRINT OCTAL RESULTS A(1,1)...A(1,(NEL+1)/2)
         EXECUTE RAP02.(NEL,NS(SN),T1,T2,LSDA,ERROR,T3)
         P'S  T1(1)...T1(50),T2(1)...T2(50),T3(1)...T3(50),ERROR
         W'R  ERROR.E.1,T'O RAP1
         EXECUTE RAP07.(NAME,S,XL,LSDA,MNAME,CC1,ERROR,NEL,CODE)
         T'H  L3,FOR J=1,1,J.G.NEL
L3       LAMBDA(SN,J)=XL(J)
         P'S  CC1...CC1(10)
         T'H  L2,FOR J=1,1,J.G.15
L2       C1(SN,J)=CC1(J)
         W'R  ERROR.E.1,T'O RAP1
         EXECUTE RAP08.(NEL,NT,S,T,ERROR,SN)
         W'R  ERROR.E.1,T'O RAP1
         PRINT COMMENT $0  CORE 1 COMPLETED$
         EXECUTE SELPGM.(2)
         E'M
```

Figure B-12. Core 1 Main Program, Program Listing (2 of 2)



CONVERT OCTAL SDA ENTRIES TO INTEGERS 1-17

INTEGERS	OCTAL	C1 FORMAT
0	00	0
1	01	1
2	02	2
3	03	3
4	04	4
5	05	5
6	06	6
7	07	7
8	10	8
9	11	9
10	21	A
11	22	B
12	23	C
13	24	D
14	25	E
15	26	F
16	27	G
17	30	H

ARGUMENTS

INPUTS	OUTPUTS
	LS LN MO DAY YR LD NEL SDA A ERROR NT TT NIT AVAIL NPL102 SN

Figure B-13. RAP01 Subprogram, Logic Flow Chart

```
        R   READ DATA TRANSFORM1,RAP01.
        R
        R   ....DEFINITION OF PROGRAM VARIABLES....
        R
        R LT(50)                WORKING ARRAY FOR CENTERING INPUT NAMES
        R MARK                  I.D. NO. IN COL. 80 OF INPUT CARDS
        R
        R   ....PROGRAM LISTING....
        R
         EXTERNAL FUNCTION(LS,LN,MO,DAY,YR,LD,NEL,SDA,A,
        1ERROR,NT,TT,NIT,AVAIL,NPL102,SN)
         ENTRY TO RAP01.
         N'S INTEGER
         F'T TT
         D'N LT(50)
         PROGRAM COMMON LZERO,RZERO
        R   IF ERROR,FIND BEGINNING OF NEXT DATA SET
        R
         W'R ERROR.E.1
LOOK      LOOK AT FORMAT FMT,MARK
          V'S FMT=$S79,I1*$
          W'R MARK.NE.1
          R'T FMT1
          V'S FMT1=$S80*$
          T'O LOOK
          E'L
          ERROR=0
         T'O RD1
         E'L
        R   READ TYPE 1 CARD
RD1      R'T DTF11,LS(1)...LS(8),SN,LN(1)...LN(46),MO,DAY,YR,MARK
         V'S DTF11=$S3,8C1,I1,S5,46C1,S5,4I3*$
         W'R MARK.NE.1,TRANSFER TO MRKERR
        R
        R   READ TYPE 2 CARD
        R
         R'T DTF12,   LD(1)...LD(46),MARK
         V'S DTF12=$S17,46C1,S16,I1*$
         W'R MARK.NE.2,TRANSFER TO MRKERR
        R
        R   LEFT JUSTIFY SYSTEM CODE
        R
         T'H LP1,FOR J=1,1,J.G.8
LP1      LT(J)=LS(J)
         I=0
         T'H LP2,FOR J=1,1,J.G.8
          W'R LT(J).NE.$ $
          I=I+1
          LS(I)=LT(J)
LP2       E'L
          W'R I.L.8
          T'H LP3,FOR J=I+1,1,J.G.8
LP3       LS(J)=$ $
          E'L
        R
        R   CENTER SYSTEM NAME
        R
         T'H LP4,FOR J=1,1,J.G.46
LP4      LT(J)=LN(J)
         WORK=1
```

Figure B-14. RAP01 Subprogram, Program Listing (1 of 3)

```
          T'O CENTER
BACK      T'H LP7,FORJJ=1,1,JJ.G.L
LP7       LN(JJ)=$ $
          JJ=JJ-1
          T'H LP8,FOR J=K+1,1,J.G.46-I
          JJ=JJ+1
LP8       LN(JJ)=LT(J)
          T'H LP9,FOR JJ=JJ+1,1,JJ.G.46
LP9       LN(JJ)=$ $
          T'O CMN
         R   CENTERING ROUTINE
CENTER    K=0
          T'H LP5,FOR J=1,1,J.G.46
          W'R LT(J).E.$ $
          K=K+1
          O'E
          T'O OUT
LP5       E'L
OUT       I=0
          T'H LP6,FOR J=46,-1,J.E.1
          W'R LT(J).E.$ $
          I=I+1
          O'E
          T'O AVG
LP6       E'L
AVG       L=(I+K)/2
          W'R WORK.E.1
          T'O BACK
          O'E
          T'O NEXT
          E'L
         R
         R   CENTER MODE NAME
         R
CMN       T'H LP10,FOR J=1,1,J.G.46
LP10      LT(J)=LD(J)
          WORK=0
          T'O CENTER
NEXT      T'H LP11,FOR JJ=1,1,JJ.G.L
LP11      LD(JJ)=$ $
          JJ=JJ-1
          T'H LP12,FOR J=K+1,1,J.G.46-I
          JJ=JJ+1
LP12      LD(JJ)=LT(J)
          T'H LP13,FOR JJ=JJ+1,1,JJ.G.46
LP13      LD(JJ)=$ $
          T'O RD3
         R
         R   CARD SEQUENCE ERROR
         R
MRKERR    P'T MERR,MARK
          V'S MERR=$1H0,I3,S2,20HCARD OUT OF SEQUENCE*$
          T'O IGNORE
IGNORE    PRINT COMMENT $0  DATA SET IGNORED$
          ERROR=1
          T'O LOOK
         R
         R   READ SYSTEM DEFINITION ARRAY(TYPE 3 CARD)
         R
RD3       R'T IN,NEL,SDA(1)...SDA(50),MARK
```

Figure B-14. RAP01 Subprogram, Program Listing (2 of 3)

```
             V'S IN=$I3,S17,50RZC1,S9,I1*$
             W'R MARK.NE.3,T'O MRKERR
            R   CHECK SDA ENTRIES AND CONVERT
            R
             W'R NEL.G.50
             PRINT COMMENT $0  NUMBER OF ELEMENTS EXCEEDS50$
             T'O IGNORE
             E'L
             T'H LP14,FOR J=1,1,J.G.NEL
             W'R SDA(J).E.0 .OR. SDA(J).E.60K .OR.SDA(J).G.30K
             T'O SDAERR
LP14         E'L
             T'H LP15,FOR J=J,1,J.G.50
             W'R SDA(J).NE.60K
             T'O SDAERR
LP15         E'L
             T'H LP16,FOR J=1,1,J.G.NEL
             W'R SDA(J).E.10K
             SDA(J)=8
             O'R   SDA(J).E.11K
             SDA(J)=9
             O'R  SDA(J).E.21K
             SDA(J)=10
             O'R  SDA(J).E.22K
             SDA(J)=11
             O'R  SDA(J).E.23K
             SDA(J)=12
             O'R  SDA(J).E.24K
             SDA(J)=13
             O'R SDA(J).E.25K
             SDA(J)=14
             O'R SDA(J).E.26K
             SDA(J)=15
             O'R SDA(J).E.27K
             SDA(J)=16
             O'R SDA(J).E.30K
             SDA(J)=17
LP16         E'L
             NPL102=(NEL+1)/2
             EXECUTE ALOAD.(NEL,SDA,A,AVAIL)
             T'O RD4
            R   ERROR IN SDA ENTRY
            R
SDAERR       PRINT COMMENT$0  ERROR IN SDA$
             T'O IGNORE
            R
            R   READ TYPE 4 CARDS (TIMES)
            R
RD4          R'T DTF14,NT,TT(1),NIT(1),TT(2),NIT(2),TT(3),NIT(3),
            1TT(4),NIT(4),TT(5),NIT(5),MARK
             V'S DTF14=$I3,5(S1,F10.2,S1,I3),S1,I1*$
             W'R MARK.NE.4,TRANSFER TO MRKERR
             T'H LP17,FOR I=6,5,I.G.NT
             R'T DTF141,TT(I),NIT(I),TT(I+1),NIT(I+1),TT(I+2),
            1NIT(I+2),TT(I+3),NIT(I+3),TT(I+4),NIT(I+4),MARK
             V'S DTF141=$S3,5(S1,F10.2,S1,I3),S1,I1*$
             W'R MARK.NE.4,TRANSFER TO MRKERR
LP17         CONTINUE
             F'N
             E'N
```

Figure B-14. RAP01 Subprogram, Program Listing (3 of 3)

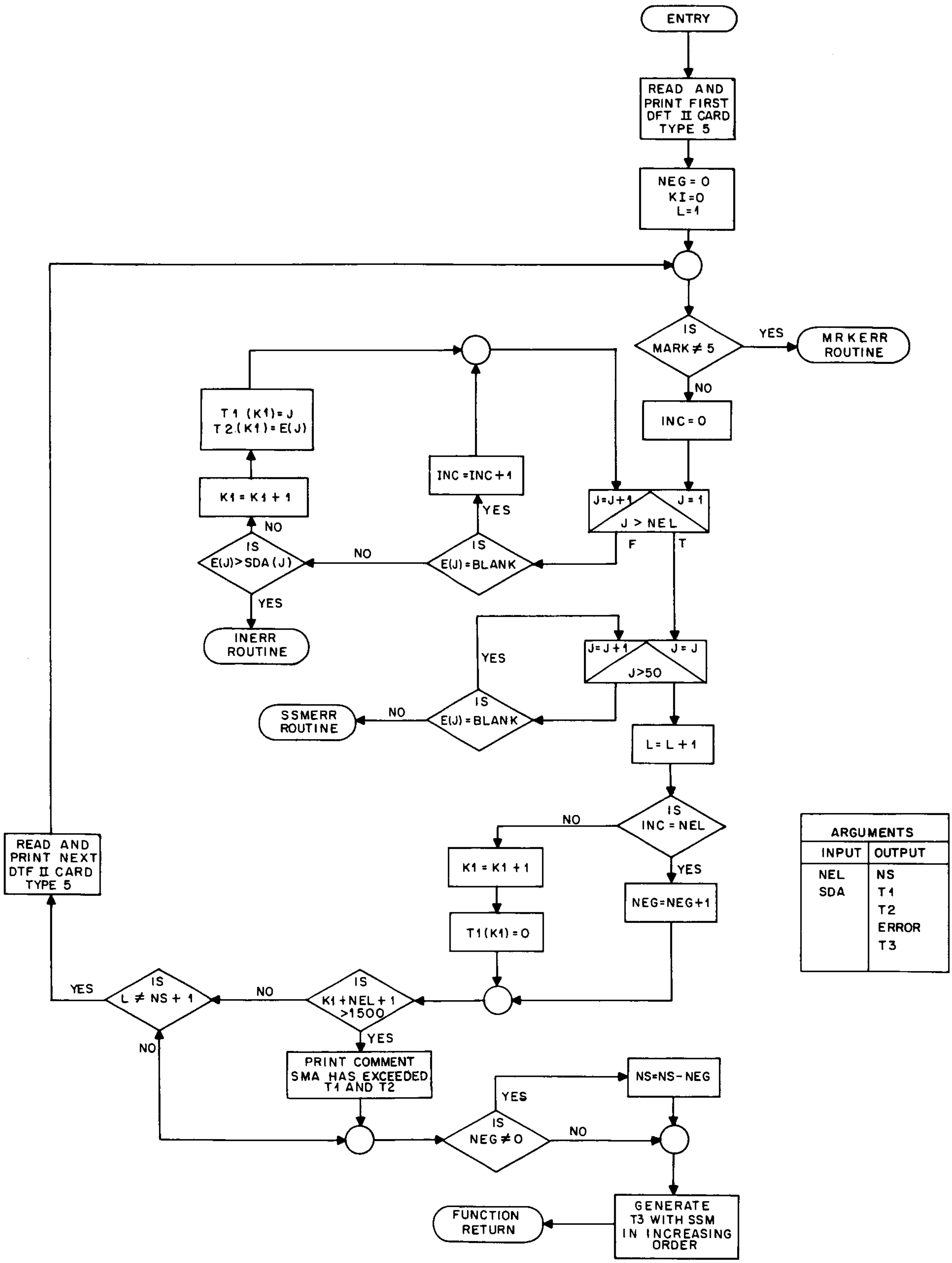


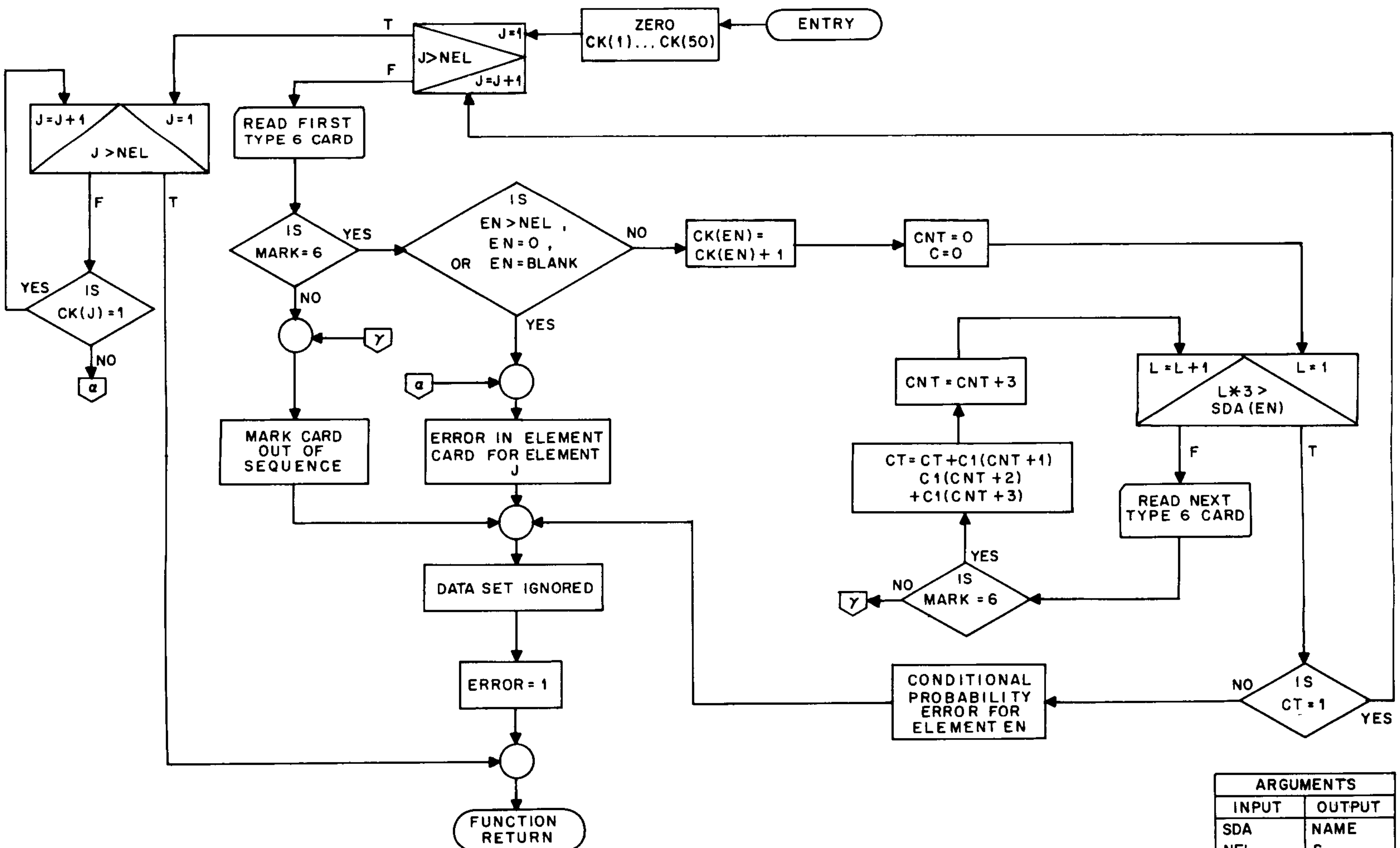
Figure B-15. RAP02 Subprogram, Logic Flow Chart

```
R  READ , CHECK ,PACK  SMA        RAP02.
R
R   ....DEFINITION OF PROGRAM VARIABLES....
R
R E(50)                   INPUT ARRAY FOR READING A ROW OF SMA
R INC                     COUNTER FOR NO. OF BLANKS IN SMA ROW
R K1                      CURRENT POSITION IN T1 AND T2
R L                       COUNTER FOR NO. OF ROWS IN SMA
R MARK                    I.D. NO. IN COL. 80 OF INPUT CARDS
R NEG                     COUNTER FOR NO. OF TOTALLY BLANK ROWS IN SMA
R T4(50)                  WORKING ARRAY FOR CONSTRUCTING T3
R
R   ....PROGRAM LISTING....
R
        EXTERNAL FUNCTION(NEL,NS,T1,T2,SDA,ERROR,T3)
        ENTRY TO RAP02.
        N'S INTEGER
        D'N E(50),T4(50)
       R  READ FIRST SSM
       R
       R
        R'T SMA1,NS,E(1)...E(50),MARK
        P'T SMA11,NS,E(1)...E(50),MARK
        V'S SMA11=$1H0,I3,S10,50I2,S9,I1*$
        V'S SMA1=$I3,S17,50I1,S9,I1*$
        L=1
        K1=0
        NEG=0
ALPHA   W'R MARK.NE.5,TRANSFER TO MRKERR
        INC=0
        T'H LP1, FOR J=1,1,J.G.NEL
        W'R E(J).E.0.AND.E(J).RS.1.NE.0
        INC=INC+1
        T'O LP1
        E'L
        W'R E(J).G.SDA(J),TRANSFER TO INERR2
        K1=K1+1
        T1(K1)=J
        T2(K1)=E(J)
LP1     CONTINUE
        T'H LP2, FOR J=J  ,1,J.G.50
        W'R E(J).E.0 .AND. E(J) .RS.1.NE.0, TRANSFER TO LP2
        T'O SSMERR
LP2     CONTINUE
        L=L+1
        W'R INC.E.NEL
        NEG=NEG+1
        T'O LOW
        E'L
        K1=K1+1
        T1(K1)=0
LOW     W'R (K1+NEL+1).G.1500,T'O XEED
        W'R L.NE.NS+1 ,TRANSFER TO NXTCRD
TEST    W'R NEG.NE.0,NS=NS-NEG
        T'O ENDRP2
       R
       R   READ NEXT SSM
       R
NXTCRD  R'T SMAN1,E(1)...E(50),MARK
        V'S SMAN1=$S20,50I1,S9,I1*$
```

Figure B-16. RAP02 Subprogram, Program Listing (1 of 2)

```
         P'T SMAN,E(1)...E(50),MARK                                         36 0229
         V'S SMAN=$1H0,S13,50I2,S9,I1*$                                     37 0230
         T'O ALPHA                                                          38 0231
        R                                                                      0232
        R     CARD  SEQUENCE ERROR                                             0233
        R                                                                      0234
MRKERR   P'T MERR,MARK                                                      39 0235
         V'S MERR=$1H0,I3,S4,20HCARD OUT OF SEQUENCE*$                         0236
         T'O IGNORE                                                         41 0237
        R                                                                      0238
        R    SSM ENTRY EXEEDS ELEMENT DEFINITION                               0239
        R                                                                      0240
INERR2   P'T SSERR,L,J                                                      44 0241
         V'S SSERR=$1H0,10HSSM NUMBER,I5,S3,33HCONTAINS ILLEGAL MODE F         0242
        1OR ELEMENT,I5*$                                                       0243
         T'O IGNORE                                                         47 0244
        R                                                                      0245
        R  ENTRY IN SMA BEYOND LEGAL NEL                                       0246
        R                                                                      0247
SSMERR   P'T SERR,L,J                                                       48 0248
         V'S SERR=$1H0,10HSSM NUMBER,I5, 37HCONTAINS A MODE FOR UNDEFI         0249
        1NED ELEMENT,I5*$                                                      0250
         T'O IGNORE                                                            0251
        R                                                                      0252
        R  SMA TOO LARGE FOR THIS PROGRAM                                      0253
        R                                                                      0254
XEED     P'T EXEED,(L-1)                                                       0255
         V'S EXEED=$1H0,67HSMA HAS EXEEDED TABLES T1 AND T2. PROGRAM W         0256
        1ILL CONTINUE USING FIRST,I3,13H  ROWS OF SMA*$                        0257
         T'O TEST                                                              0258
IGNORE   PRINT COMMENT$0   DATA SET IGNORED$                                   0259
         ERROR=1                                                               0260
        R  BUILD T3 TABLE FOR SSM'S IN INCREASING ORDERS                       0271
        R                                                                      0272
ENDRP2   JJ=0                                                                  0273
         ORDER=0                                                               0274
         OT=1                                                                  0275
GAMMA    KK=0                                                                  0276
         T'H DELTA,FOR JJ=JJ+1,1,T1(JJ).E.0                                    0277
DELTA    KK=KK+1                                                               0278
         T4(OT)=KK                                                             0279
         W'R OT.NE.NS                                                          0280
         OT=OT+1                                                               0281
         T'O GAMMA                                                             0282
         E'L                                                                   0283
         KK=0                                                                  0284
ZETA     II=1                                                                  0285
         ORDER=ORDER+1                                                         0286
         T'H SIGMA,FOR JJ=1,1,JJ.G.NS                                          0287
         W'R T4(JJ).E.ORDER                                                    0288
         KK=KK+1                                                               0289
         T3(KK)=II                                                             0290
         E'L                                                                   0291
SIGMA    II=II+T4(JJ)+1                                                        0292
         W'R KK.NE.NS,T'O ZETA                                                 0293
         KK=KK+1                                                               0294
         T3(KK)=0                                                              0295
         F'N                                                                   0326
         E'N                                                                   0327
```

Figure B-16. RAP02 Subprogram, Program Listing (2 of 2)



ARGUMENTS	
INPUT	OUTPUT
SDA	NAME
NEL	S
	LAMBDA
	MNAME
	C1
	ERROR CODE

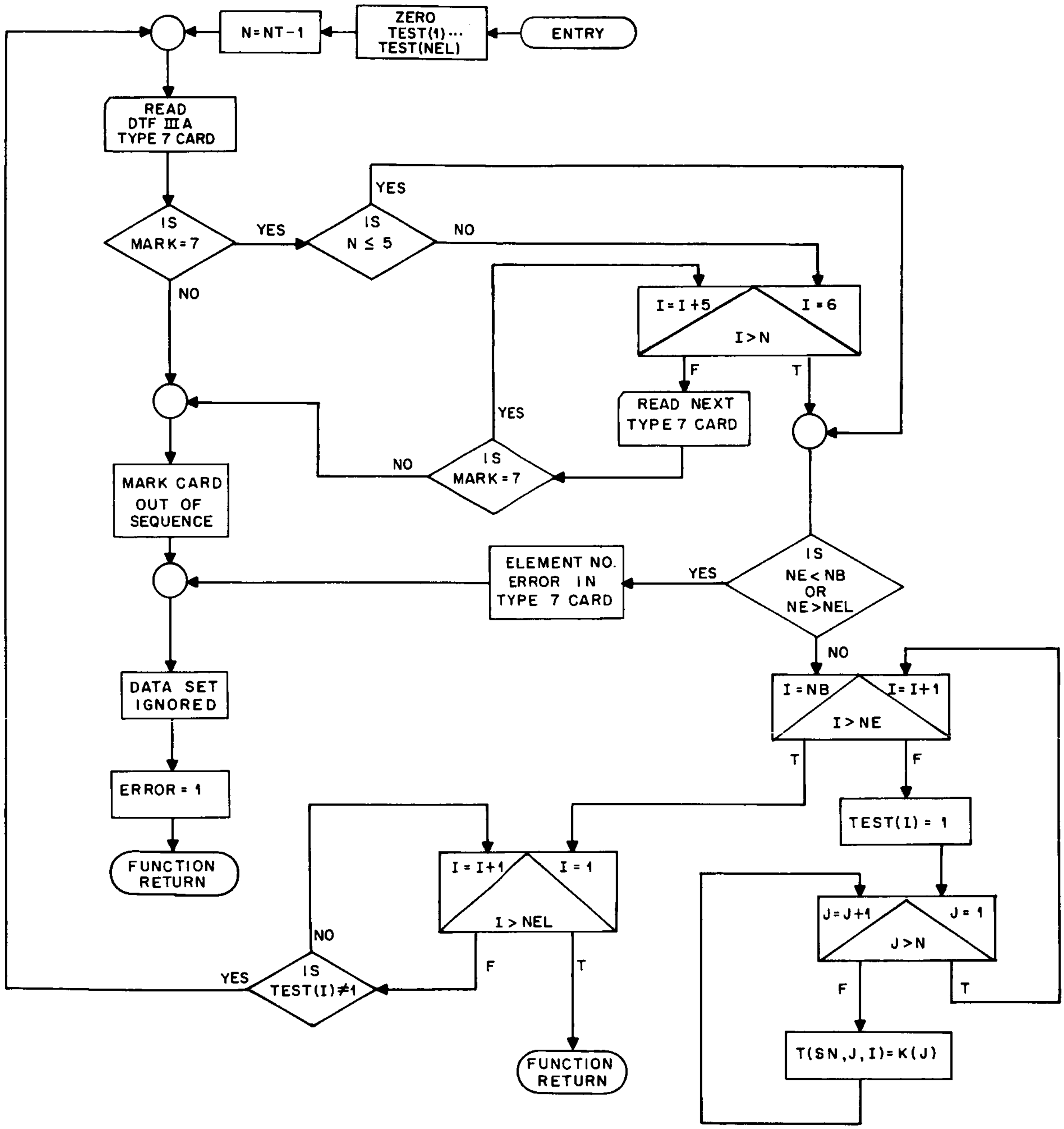
Figure B-17. RAP07 Subprogram, Logic Flow Chart

```
R  READ DATA TRANS FORM 3 .RAP07.                                        0766
R                                                                         0767
R    ....DEFINITION OF PROGRAM VARIABLES....
R
R ACM                      COUNTER FOR SEQUENTIAL POSITION IN C1
R CC1(17)                  WORKING ARRAY FOR READING MODES FOR INDIV.
R                          ELEMENTS
R CK(50)                   CHECKING ARRAY FOR READING DATA ON EACH
R                          ELEMENT
R CNT                      COUNTER FOR SEQUENTIAL POSITION IN CC1
R CT                       LOCATION FOR TOTALLING COND. PROB.
R EN                       CURRENT ELEMENT NO.
R MARK                     SEQUENCE CODE FOR INPUT CARDS
R
R    ....PROGRAM LISTING....
R
 EXTERNAL FUNCTION(NAME,S,LAMBDA,SDA,MNAME,C1,ERROR,NEL,CODE)
 ENTRY TO RAP07.                                                          0769
 D'N CK(50),CC1(17)
 N'S INTEGER                                                              0771
 FLOATING POINT CT,LAMBDA,C1,CC1                                          0772
R                                                                         0773
R   READ DATA TRANS FORM 3 (TYPE 6 CARD)                                  0774
R                                                                         0775
 EXECUTE ZERO.(CK(1)...CK(50))                                            0776
 TOT=0                                                                    0777
 ACM=0                                                                    0778
 T'H LP1,FOR J=1,1,J.G.NEL                                                0779
 R'T DTF31,EN,NAME(EN,1),NAME(EN,2),S(EN),LAMBDA(EN),                     0780
1CODE(EN),MARK
 V'S DTF31=$S1,I2   ,S1,C6, C4,S1,I1,S1,F10.2,S2,                          0782
1C6,S44,I1*$
 W'R MARK.NE.6,TRANSFER TO MRKERR                                         0784
 W'R EN.G.NEL.OR.EN.E.0.AND.EN.RS.1.NE.0.OR.EN.E.0                        0785
 P'S EN                                                                   0786
 T'O ELNERR                                                               0787
 E'L                                                                      0788
 CK(EN)=CK(EN)+1                                                          0790
 P'S CK(EN)                                                               0791
R                                                                         0792
R  READ TYPE 6 CARDS (MODES)                                              0793
R                                                                         0794
 CNT=0                                                                    0795
 CT=0.                                                                    0796
 T'H LP3,FOR L=1,1,CNT.GE.SDA(EN)                                         0797
 P'S CNT                                                                  0798
 R'T DTF32,MNAME(ACM+1,1)...MNAME(ACM+1,3),C1(ACM+1),                     0799
1MNAME(ACM+2,1)...MNAME(ACM+2,3),C1(ACM+2),MNAME(ACM+3,1)...              0800
2MNAME(ACM+3,3),C1(ACM+3),MARK                                            0801
 CC1(CNT+1)=C1(ACM+1)                                                     0802
 CC1(CNT+2)=C1(ACM+2)                                                     0803
 CC1(CNT+3)=C1(ACM+3)                                                     0804
 W'R MARK.NE.6,TRANSFER TO MRKERR                                         0805
 V'S DTF32=$S4,3(S2,2 C6, C3,S1,F6.2,S1),I1*$                             0806
 P'S CT,ACM                                                               0807
 W'R CNT+3.G.SDA(EN)                                                      0808
  W'R CNT+2.G.SDA(EN)                                                     0809
 CC1(CNT+2)=0.                                                            0810
  E'L                                                                     0811
 CC1(CNT+3)=0.                                                            0812
```

Figure B-18. RAP07 Subprogram, Program Listing (1 of 2)

```
          E'L
          CT=CT+CC1(CNT+1)+CC1(CNT+2)+CC1(CNT+3)
          P'S CT
          CNT=CNT+3
LP3       ACM=ACM+3
          W'R.ABS.(CT-1.).G. .00001,T'O CERRR
          TOT=TOT+SDA(EN)
          ACM=TOT
LP1       CONTINUE
          T'H LP4,FOR J=1,1,J.G.NEL
LP4       W'R CK(J).NE.1,TRANSFER TO ELERR
          T'O ENDRP7
R
R           ERROR IN ELEMENT CARD
R
ELERR     P'T ECER,J
          V'S ECER=$1H0,S2,33HERROR IN ELEMENT CARD FOR ELEMENT,I3*$
          T'O IGNORE
R
R           CONDITIONAL PROBABILITY ERROR
R
CERRR     P'T CERR,EN
          V'S CERR=$1H0,42HCONDITIONAL PROBABILITY ERROR FOR ELEMENT ,
         1I3*$
          T'O IGNORE
R
R           ILLEGAL S IN TYPE 6 CARD
R
SERR      PRINT COMMENT$0  INCORRECT SYMBOL S IN TYPE 6 CARD$
          T'O IGNORE
R
R           ELEMENT NBR ERROR
R
ELNERR    PRINT COMMENT$0  ELEMENT NUMBER ERROR IN TYPE 6 CARD$
          T'O IGNORE
R
R           CARD OUT OF SEQUENCE
R
MRKERR    P'T MERR,MARK
          V'S MERR=$1H0,I3,S2,20HCARD OUT OF SEQUENCE*$
IGNORE    PRINT COMMENT$0  DATA SET IGNORED$
          ERROR=1
ENDRP7    F'N
          E'N
```

Figure B-18. RAP07 Subprogram, Program Listing (2 of 2)



ARGUMENTS	
INPUT	OUTPUT
NEL NT S SN	T ERROR

Figure B-19. RAP08 Subprogram, Logic Flow Chart

```
R          READ K FACTORS            RAP08.          (PRESTO)
R                                                                       0859
R     ....DEFINITION OF PROGRAM VARIABLES....
R
R K(10)                     WORKING ARRAY FOR READING K FACTORS
R MARK                      SEQUENCE CODE FOR INPUT CARDS
R N                         NO. OF TRANSITION TIMES MINUS 1
R TEST(50)                  CHECKING ARRAY TO ASSURE K'S ARE READ FOR
R                           EVERY ELEMENT
R
R     ....PROGRAM LISTING....
R
        EXTERNAL FUNCTION(NEL,NT,S,T,ERROR,SN)
        ENTRY TO RAP08.                                                 0861
        F'T K,T
        N'S INTEGER                                                     0863
        D'N TEST(50),K(10)
        T'H ALPHA,FOR I=1,1,I .G.NEL                                    0865
ALPHA   TEST(I)=0                                                       0866
        N=NT-1
DELTA   READ FORMAT F1,NB,NE,K(1)...K(5),MARK                           0872
        V'S F1=$S1,I2,S2,I2,5(S2,F10.2,S2),I3*$                         0873
        W'R MARK.NE.7,TRANSFER TO ER1                                   0874
        W'R N.LE.5,TRANSFER TO ZETA                                     0875
        T'H EPS,FOR I=6,5,I.G.N                                         0876
        READ FORMAT F2,K(I),K(I+1),K(I+2),K(I+3),K(I+4),MARK            0877
        V'S F2=$S7,5(S2,E10.2,S2),I3*$                                  0878
        W'R MARK.NE.7,TRANSFER TO ER1                                   0879
EPS     CONTINUE                                                        0880
ZETA    W'R NE.L.NB.OR.NE.G.NEL,TRANSFER TO ER2                         0881
        T'H ETA,FOR I=NB,1,I.G.NE                                       0882
        TEST(I)=1                                                       0883
        T'H ETA,FOR J=1,1,J.G.N                                         0884
ETA     T(SN,J,I)=K(J)
        T'H THETA,FOR I=1,1,I.G.NEL                                     0886
        W'R                TEST(I).NE.1,TRANSFER TO DELTA
THETA   CONTINUE                                                        0888
        F'N                                                             0892
ER1     PRINT FORMAT F3,MARK                                            0893
        V'S F3=$1H0,I3,S3,20HCARD OUT OF SEQUENCE*$                     0894
        T'O IGNORE                                                      0895
ER2     PRINT COMMENT$0  ELEMENT NUMBER ERROR IN TYPE 7 CARD$           0896
        T'O IGNORE                                                      0897
IGNORE  PRINT COMMENT$0  DATA SET IGNORED$                              0898
        ERROR=1                                                         0899
        F'N                                                             0900
        E'N                                                             0901
```

Figure B-20. RAP08 Subprogram, Program Listing

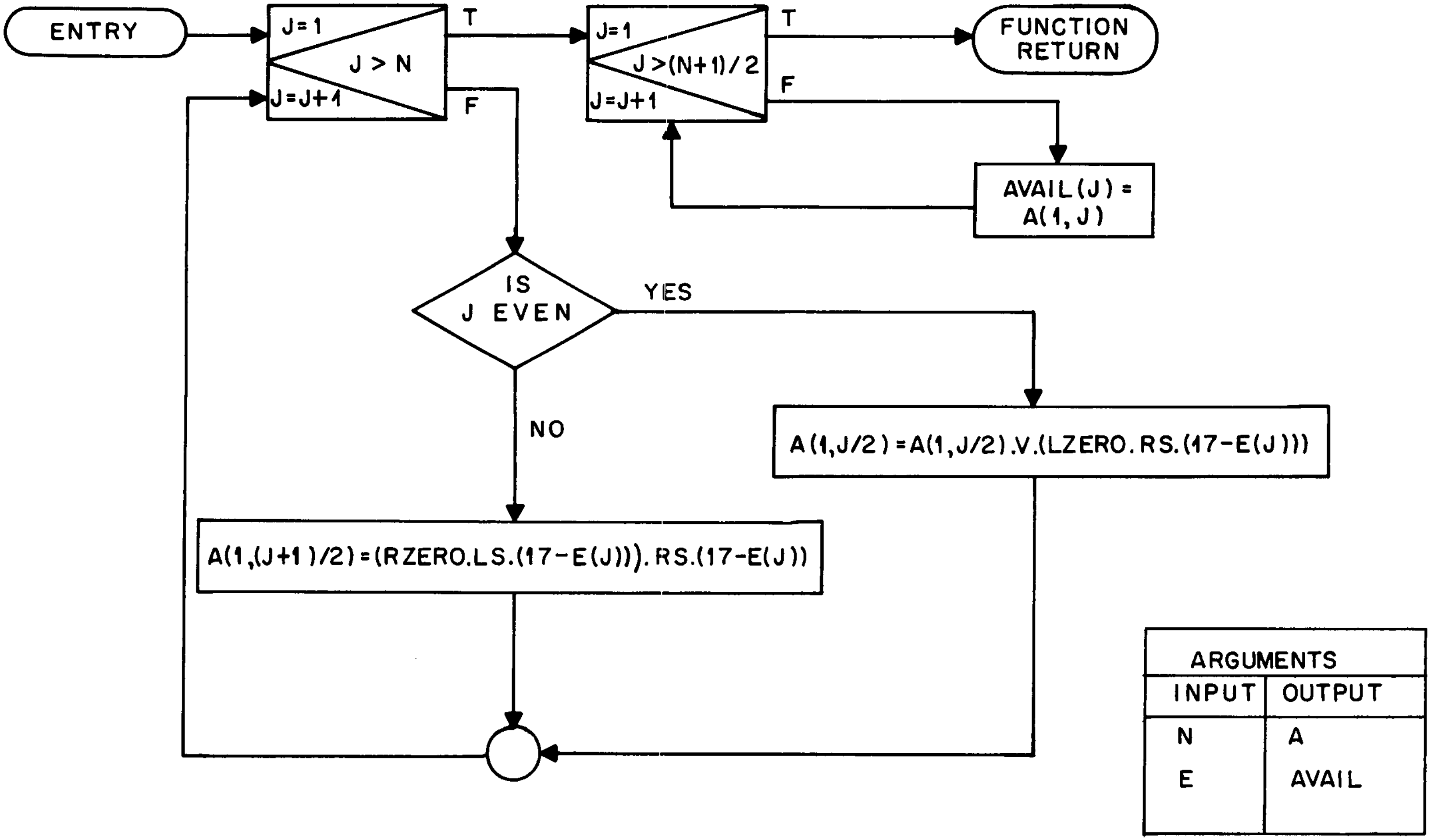


Figure B-21. ALOAD Subprogram, Logic Flow Chart

```
R     SUBROUTINE   ALOAD.                                                  0329
R           PLACE   ELEMENTS  INTO  HALFWORDS  FROM  SDA                   0330
R
R     ....DEFINITION OF PROGRAM VARIABLES....
R
R A                          DOUBLE SUBSCRIPTED OUTPUT ARRAY
R AVAIL                      SINGLE SUBSCRIPTED OUTPUT ARRAY
R E                          INPUT ARRAY DESIGNATING NO. OF 1 BITS TO GO
R                            INTO WORD
R LZERO                      SYMBOLIC WORD WITH ZEROES IN LEFT HALF AND 1
R                            BITS IN RT HALF
R N                          NUMBER OF HALF-WORDS TO BE GENERATED
R RZERO                      SYMBOLIC WORD WITH ZEROES IN RT HALF AND 1
R                            BITS IN LEFT HALF
R
R     ....PROGRAM LISTING....
R
       EXTERNAL FUNCTION(N,E,A,AVAIL)                                         0331
       ENTRY TO ALOAD.                                                    02  0332
       N'S INTEGER                                                        03  0333
       PROGRAM COMMON LZERO,RZERO                                         04  0334
       T'H END, FOR J=1,1,J.G.N                                           05  0335
         W'R(J/2)*2.E.J                                                   06  0336
         A(1,J/2)=A(1,J/2).V.(LZERO.RS.(17-E(J)))                         07  0337
         O'E                                                              08  0338
         A(1,(J+1)/2)=(RZERO.LS.(17-E(J))).RS.(17-E(J))                   09  0339
END      E'L                                                              10  0340
       T'H ONE, FOR J=1,1,J.G.(N+1)/2                                     11  0341
ONE    AVAIL(J)=A(1,J)                                                        0342
       F'N                                                                13  0343
       E'N                                                                14  0344
```

Figure B-22. ALOAD Subprogram, Program Listing

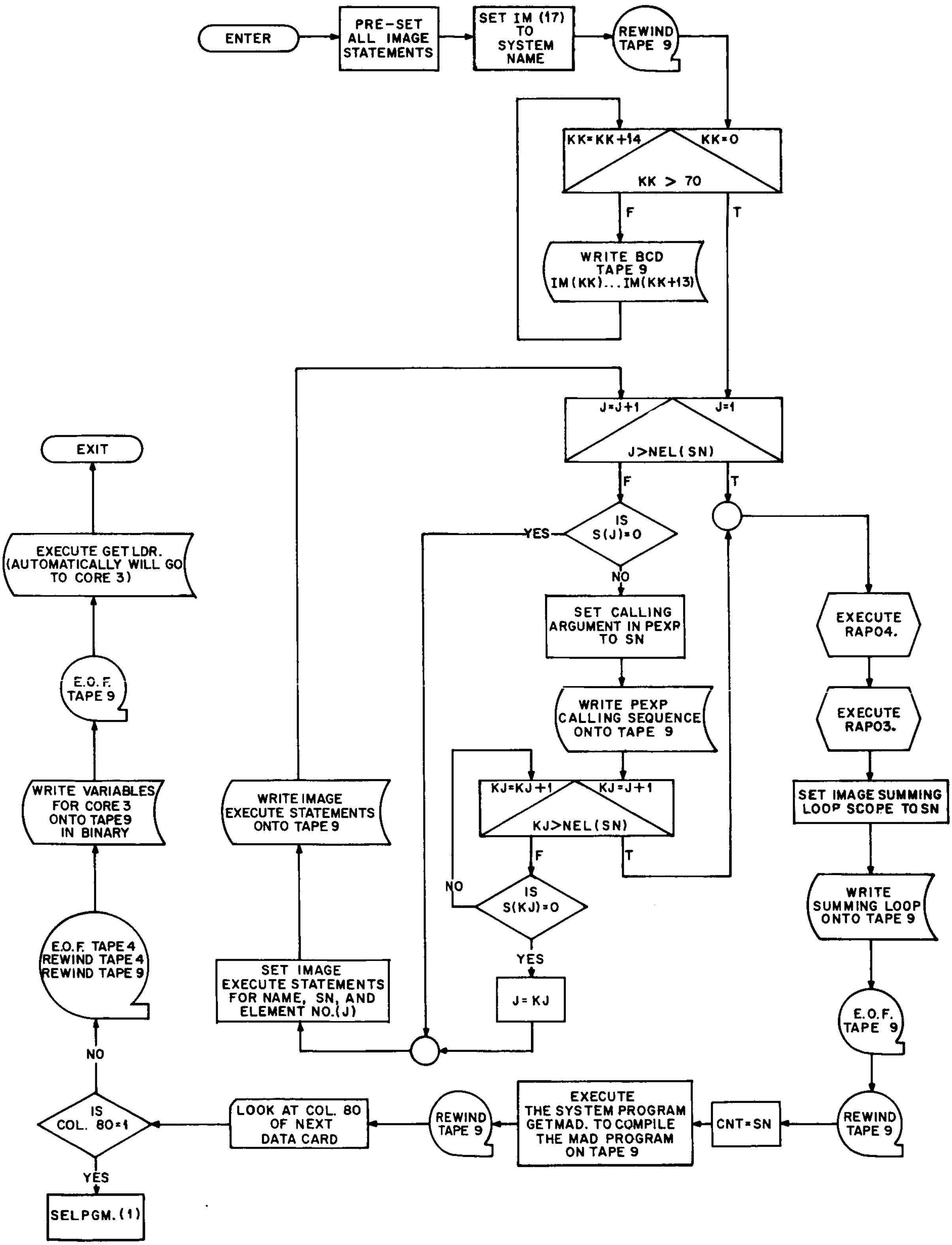


Figure B-23. Core 2 Main Program, Logic Flow Chart

```
R   MAIN PROGRAM FOR RAPID CORE 2
R
R   ....DEFINITION OF PROGRAM VARIABLES....
R
R A(1000,ADIM)            WORKING ARRAY TO STORE RESIDUES DURING STRIP
R ADIM(2)                 DIMENSION VECTOR FOR A.
R AVAIL(25)               ORIGINAL SYMBOLIC SDA FOR CURRENT SYSTEM
R C1(10*15)               COND. PROB. LIST BY SYSTEM NO.
R CODE(15)                EL. CODE NAMES - CURRENT SYSTEM (6 BCD CHAR.)
R IM(270)                 IMAGE ARRAY FOR OPENING AND CLOSING STATEMENTS
R                         OF GEN. PROGRAM
R KIM(14)                 IMAGE ARRAY FOR CALLING PEXP IN GEN. PROG.
R LAMBDA(150,LDIM)        FAILURE RATES BY SYS. NO. AND EL.
R LDIM(2)                 DIMENSION VECTOR FOR LAMBDA - USES SET. SUBR.
R LD(46)                  CURRENT SYSTEM MODE NAME (CENTERED)
R LIM(61)                 IMAGE ARRAY FOR CALLING STATEMENTS OF GEN.PROG
R LN(46)                  CURRENT SYSTEM NAME (CENTERED)
R LS(8)                   CURRENT SYSTEM CODE NAME
R LSDA(50)                CURRENT SYSTEM SDA - NOT SYMBOLIC
R LZERO                   SYMBOLIC WORD WITH ZEROES IN LEFT HALF AND 1
R                         BITS IN RT HALF
R NEL(30)                 NO. OF ELEMENTS BY SYSTEM NO. (NEL(0)= FOR
R                         CURRENT SYS.)
R NIT(10)                 NO. OF INTERNAL TIMES BETWEEN N AND (N+1) ST
R NS(10)                  NO. OF SUBMODES BY SYSTEM NO.
R RZERO                   SYMBOLIC WORD WITH ZEROES IN RT HALF AND 1
R                         BITS IN LEFT HALF
R S(15)                   S CODE FOR ELEMENTS IN CURRENT SYSTEM
R SDA(10*15)              SDA BY SYSTEM AND ELEMENT NO. - NOT SYMBOLIC
R SORT(1000,ADIM)         WORKING ARRAY FOR STRIP PROCESS IN RAP03.
R T(10*10*15)             K FACTORS BY SYS. NO., TIME INT., AND EL. NO.
R T1(50)                  ELEMENT DESIGNATION FOR SMA SHORT FORM
R T2(50)                  MODE NO. FOR THE EL. OF IDEN. POSITION IN T1 -
R                         SMA SHORT FORM
R T3(50)                  ORGANIZATION OF SMA BY INC. ORDER OF ROWS -
R                         INDICATES POSITION IN T1 WHERE ROW BEGINS.
R TT(10)                  TRANSITION TIMES
R XIPT(22)                FAILURE RATES OF SYSTEM ELEMENTS BY EL. NO.
R XL(15)                  CURRENT SYSTEM ELEMENT FAILURE RATES
R
R   ....PROGRAM LISTING....
R
 N'R
 F'T TT,LAMBDA,C1,T,CC1
 V'S ADIM=2,1,0
 V'S LDIM=SET.,0,0
 P'N LZERO,RZERO,A(1000,ADIM),ADIM,T1(50),T2(50),T3(50),
1AVAIL(25),SDA(10*15),NS(10),NEL(30),ERROR,LS(8),LN(46),LD(46)
2,TT(10),NIT(10),S(15),LAMBDA(150,LDIM),ZZZZ,NT,SN,CODE(15),C1
3(10*15),T(10*10*15),LSDA(50),MO,DAY,YR
 V'S ZZZZ = 0,0,0
 D'N SORT(1000,ADIM)
 D'N  XIPT(22)
 EQUIVALENCE (XIPT,LAMBDA),(XDIM,LDIM)
R
R   FIRST PART OF IMAGE PROGRAM
R
 V'S IM(0 )=$             EXTERNAL FUNCTION(SUMTP)
1                                   $
 V'S IM(14)=$             E'O          .
```

Figure B-24. Core 2 Main Program, Program Listing (1 of 3)

```
             1                                    $
              V'S IM(28)=$            D'N BP(100),TP(100),P(30*5),PP(30)
             1                                    $
              V'S IM(42)=$            INTEGER J,NEL,NS,NT,SDA,SET,
             1                                    $
              V'S IM(56)=$            P'N NEL(10),NS(10),NT,TT(10),T(10*10*1
             15),XIPT(150),                       $
              V'S IM(70)=$           1C1(10*15),SDA(10*15),R(10*15)
             1                                    $
             R
             R   SECOND PART OF IMAGE PROGRAM
             R
              V'S LIM(0  )=$            EXECUTE              .(SUMTP)
             1                                     $
              V'S LIM(14)=$             P(             ,1 )=SUMTP
             1                                     $
              V'S LIM(28)=$             PP(            )=1.-SUMTP
             1                                     $
              V'S LIM(42)=$             R(             ,            )=1.-SUMTP
             1                                     $
             R
             R   THIRD PART OF IMAGE PROGRAM
             R
              V'S IM(200)=$             SUMTP=0.
             1                                     $
              V'S IM(214)=$             T'H LA,FOR J=1,1,J.G.NS(          )
             1                                     $
              V'S IM(228)=$LA           SUMTP=SUMTP+TP(J)
             1                                     $
              V'S IM(242)=$             F'N
             1                                     $
              V'S IM(256)=$             E'N
             1                                     $
              V'S KIM(0  )=$            EXECUTE PEXP.(P,PP,         )
             1                                     $
              IM(17)=COMPZ.(6,LS(1))
              REWIND TAPE 9
             R
             R   WRITE FIRST PART OF IMAGE ONTO TAPE
             R
              T'H WWR,FOR KK=0,14,KK.G.70
WWR           WRITE BCD TAPE 9,FM,IM(KK)...IM(KK+13)
              T'H LA,FOR J=1,1,J.G.NEL(SN)
              W'R S(J).E.0
BG            LIM(4)=CODE(J)
              LIM(17)=BNBCD.(J)
              LIM(31)=BNBCD.(J)
              LIM(45)=BNBCD.(SN)
              LIM(47)=LIM(31)
             R
             R  WRITE LIM(0) TO LIM(41) ONTO TAPE 9
             R
              T'H WR,FOR K=0,14,K.G.42
WR            WRITE BCD TAPE 9,FM,LIM(K)...LIM(K+13)
              V'S FM=$14C6*$
              T'O LA
              E'L
              KIM(5)=BNBCD.(SN)
              WRITE BCD TAPE 9,FM,KIM(0)...KIM(13)
              T'H LLA,FOR KJ=J+1,1,KJ.G.NEL(SN)
```

Figure B-24. Core 2 Main Program, Program Listing (2 of 3)

```
            W'R S(KJ).E.0
            J=KJ
            T'O BG
            E'L
LLA         CONTINUE
             T'O EXR4
LA          CONTINUE
EXR4        EXECUTE RAP04.(NS(SN),T1,T2,T3)
            EXECUTE RAP03.(T1,T2,T3,SORT,A,NS(SN),NEL,LSDA,AVAIL)
            IM(220)=BNBCD.(SN)
            T'H LB,FOR J=200,14,J.G.269
LB          WRITE BCD TAPE 9,FM,IM(J   )...IM(J+13)
            END OF FILE TAPE 9
            REWIND TAPE 9
            CNT=SN
            PRINT RESULTS CNT
            PRINT COMMENT $0    ... GOING AFTER MAD ...$
            GETMAD.(NEXT,ER,9,4,10,11,3,0,1)
 NEXT       CONTINUE
            REWIND TAPE 9
            PRINT COMMENT $1$
            LOOK AT FORMAT FMT,MARK
            V'S FMT=$S79,I1*$
            W'R MARK.E.1,EXECUTE SELPGM.(1)
            END OF FILE TAPE 4
            REWIND TAPE 4
            REWIND TAPE 9
            WRITE BINARY TAPE 9,NEL...NEL(10),NT,SN,NS ...NS(10)
            WRITE BINARY TAPE9,T(1,1,1)...T(SN,NT,NEL(SN)),XIPT(1)...XIPT
           1(22),      TT...TT(NT),LD...LD(46),LS...LS(8),SDA(1,1)...SDA(S
           2N,15),C1(1,1)...C1(SN,15),MO,DAY,YR
            END OF FILE TAPE 9
            EXECUTE GETLDR.(0,0,2,3,4,1,0,0,0,0,0,0,10,000000,2,1,0,0)
ER          PRINT COMMENT $0  ERROR IN GETTING MAD, CHECK OUT TERMINATED$
            SELPGM.(1)
            E'M
```

Figure B-24. Core 2 Main Program, Program Listing (3 of 3)

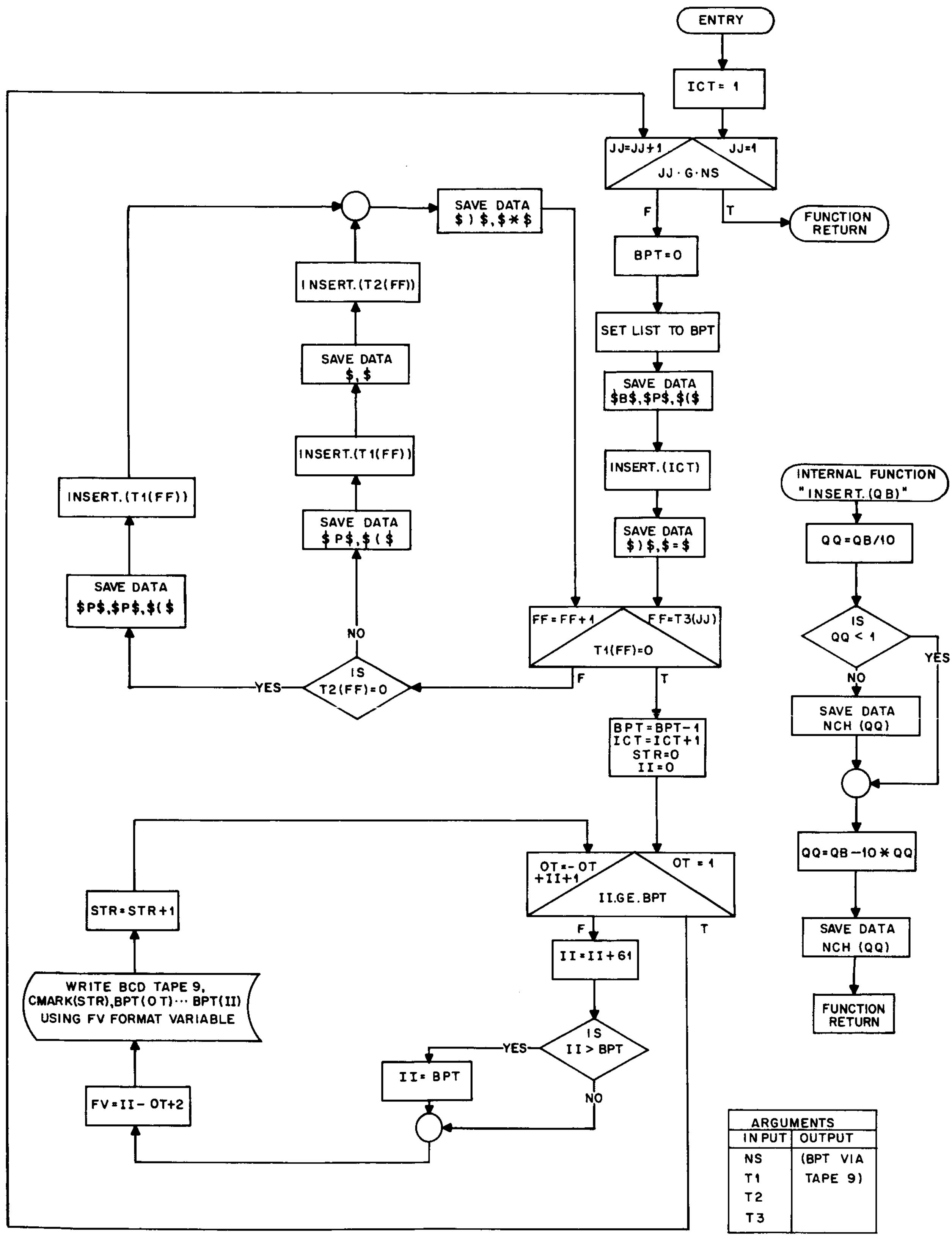


Figure B-25. RAP04 Subprogram, Logic Flow Chart

```
R  BP EQUATION GENERATOR AND LOADER---  RAP04.
R
R   ....DEFINITION OF PROGRAM VARIABLES....
R
R BPT(600)                ARRAY FOR BLDG. EACH BASIC PROB. EQ'N.
R CMARK(10)               PRESET ARRAY OF BCD CHARACTERS
R ICT                     COUNTER FOR CURRENT SUBSCRIPT FOR BP EQN.
R II                      COUNTER FOR LENGTH OF CARD IMAGE
R NCH(10)                 PRESET ARRAY OF BCD CHARACTERS
R STR                     SUBSCRIPT OF CMARK
R
R   ....PROGRAM LISTING....
         EXTERNAL FUNCTION(NS,T1,T2,T3)
         E'O RAP04.
         N'S INTEGER
         FORMAT VARIABLE FV
         D'N BPT(600),NCH(10)
         V'S CMARK=$ $,$1$,$2$,$3$,$4$,$5$,$6$,$7$,$8$,$9$
         V'S NCH=$0$,$1$,$2$,$3$,$4$,$5$,$6$,$7$,$8$,$9$
         ICT=1
         INTERNAL FUNCTION(QB)
         E'O INSERT.
         QQ=QB/10
         W'R QQ.L.1,T'O QQON
         SAVE DATA NCH(QQ)
QQON     QQ=QB-10*QQ
         SAVE DATA NCH(QQ)
         F'N
         E'N
         T'H ANOTHR,FOR JJ=1,1,JJ.G.NS
         BPT=0
         SET LIST TO BPT
         SAVE DATA $B$,$P$,$($
         INSERT.(ICT)
         SAVE DATA $)$,$=$
         T'H LEMON,FOR FF=T3(JJ),1,T1(FF).E.0
         W'R T2(FF).E.0
         SAVE DATA $P$,$P$,$($
         INSERT.(T1(FF))
         O'E
         SAVE DATA $P$,$($
         INSERT.(T1(FF))
         SAVE DATA $,$
         INSERT.(T2(FF))
         E'L
LEMON    SAVE DATA $)$,$*$
         BPT=BPT-1
         ICT=ICT+1
         STR=0
         II=0
         T'H ORANGE, FOR OT=1,-OT+II+1,II.GE.BPT
         II=II+61
         W'R II.G.BPT,II=BPT
         FV=II-OT+2
         WRITE BCD TAPE 9,FMT,CMARK(STR),BPT(OT),...,BPT(II)
         V'S FMT=$S10,'FV'C1,T80*$
ORANGE   STR=STR+1
ANOTHR   CONTINUE
         F'N
         E'N
```

Figure B-26. RAP04 Subprogram, Program Listing

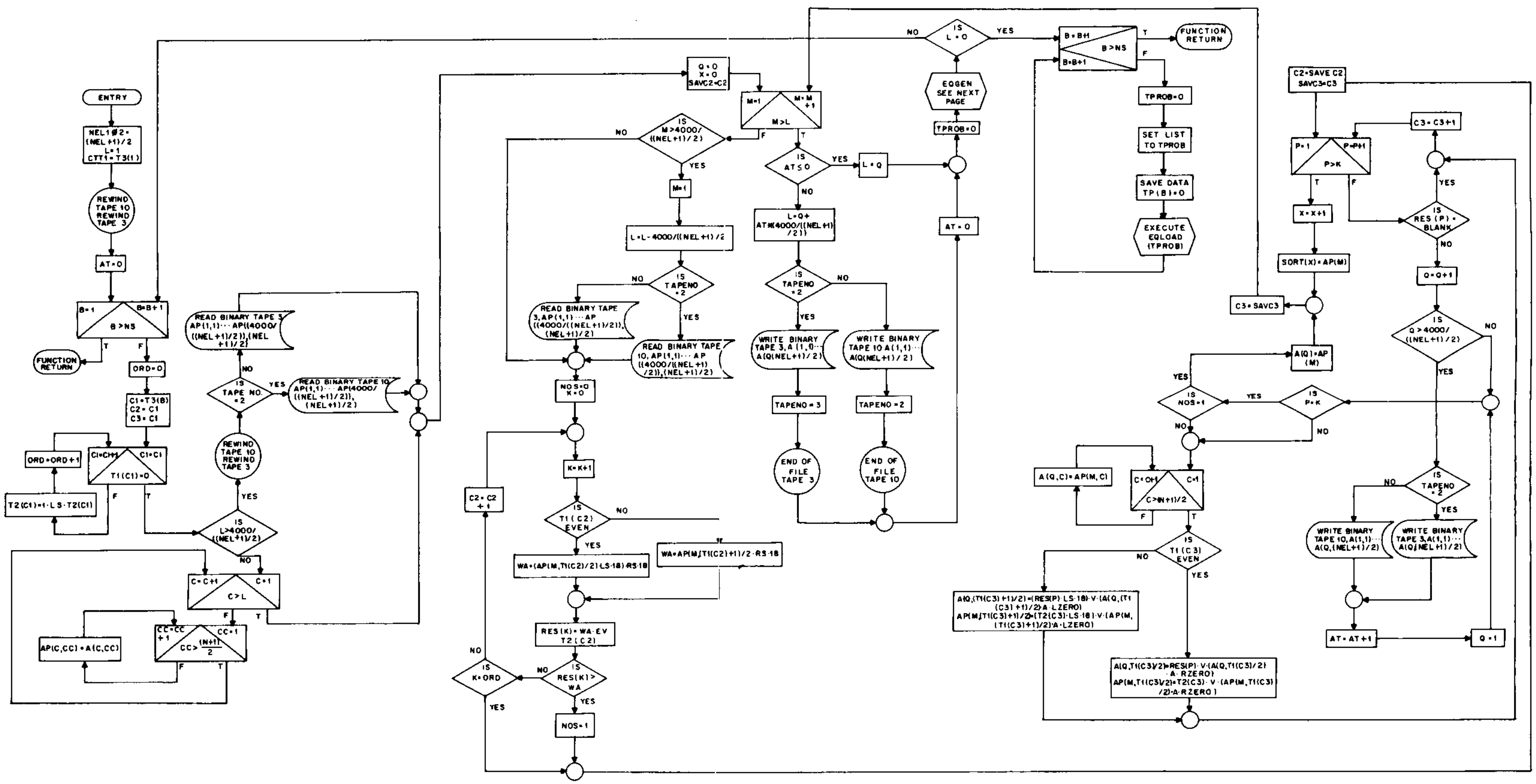


Figure B-27. RAP03 Subprogram, Logic Flow Chart (1 of 2)

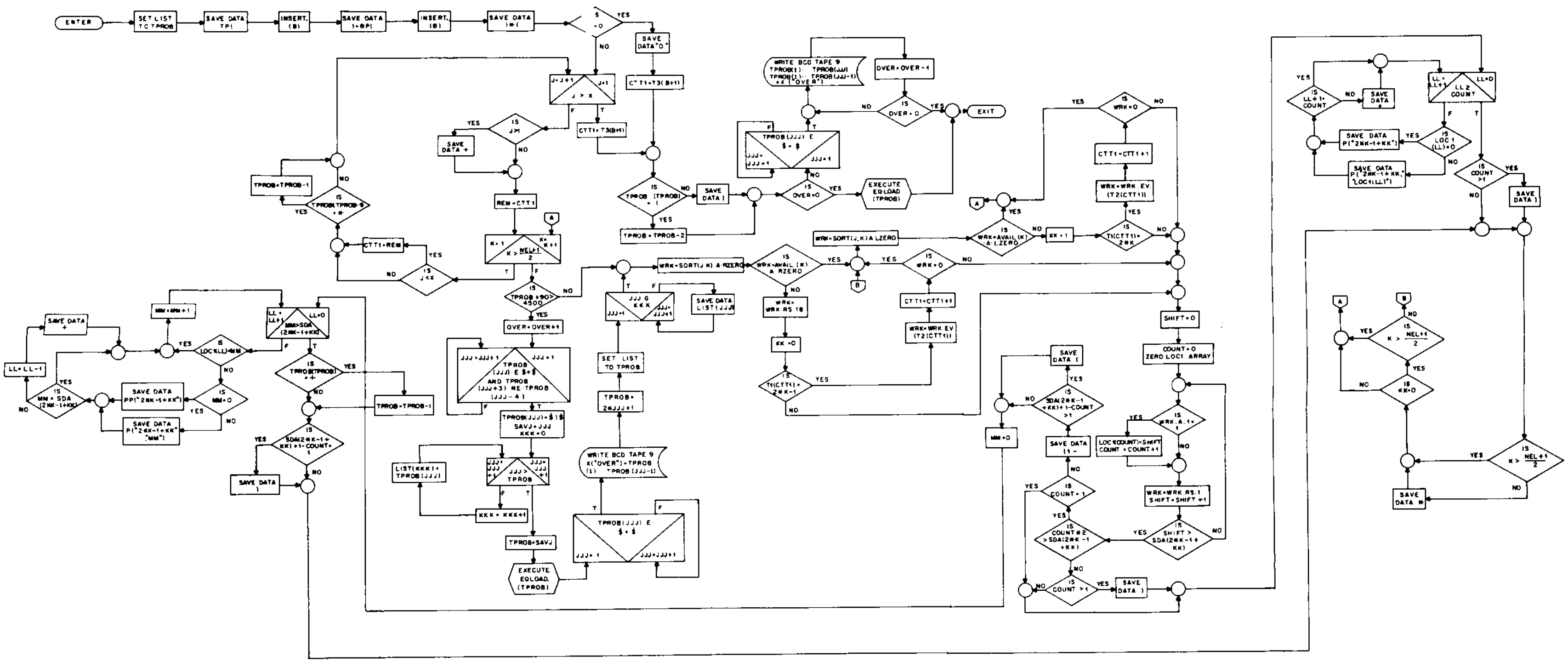


Figure B-27. RAP03 Subprogram, Logic Flow Chart (2 of 2)

```
R     STRIP  LOGIC   RAP03.
R
R    ....DEFINITION OF PROGRAM VARIABLES....
R
R AP(1000,APDIM)          WORKING SYMBOLIC ELEMENT ARRAY FOR STRIP LOGIC
R APDIM(2)                DIMENSION VECTOR FOR AP
R AT                      NO. OF OVERFLOWS
R C1                      POSITIONS IN T2 ARRAY
R C2                      POSITIONS IN T2 ARRAY
R C3                      POSITIONS IN T1 ARRAY
R COUNT                   COUNTER FOR NO. OF 1 BITS IN SHIFT OPERATION
R                         ON WRK
R CTT1                    COUNTER FOR POSITION IN T2
R L                       COUNTER FOR NO. OF CURRENT RESIDUES
R LIST                    WORKING ARRAY TO SAVE LAST SECTION OF TPROB
R LOC1(17)                ARRAY FOR DEFINING POSITION OF 1 BIT IN EQN.
R                         BUCKET
R NOS                     SWITCH TO DETERMINE IF SORT IS EMPTY
R ORD                     ORDER OF CURRENT ROW OF SMA
R OVER                    INDICATING FOR TPROB OVERFLOW
R Q                       SUBSCRIPT ON NO. OF -A- POSITIONS AVAILABLE
R RES(50)                 RESULT ARRAY OF EXCLUSIVE OR OPERATION IN
R                         STRIP
R SAVE2                   LOCATION TO SAVE ORIGINAL C2 VALUE
R SAVC3                   LOCATION TO SAVE ORIGINAL C3 VALUE
R SHIFT                   COUNTER FOR NO. OF SHIFTS PERFORMED ON WRK
R SORT(1000,ADIM)         WORKING ARRAY FOR STRIP PROCESS
R TPROB(4500)             ARRAY FOR BLDG. EACH TOTAL ROW PROB EQN.
R WRK                     WORKING SYMBOLIC ARRAY FOR EQN. GENERATION
R X                       SUBSCRIPT ON NO. OF SORT POSITIONS AVAILABLE
R
R    ....PROGRAM LISTING....
R
R
        EXTERNAL FUNCTION (T1,T2,T3,SORT,A,NS,NEL,SDA,AVAIL)
        ENTRY TO RAP03.
        N'S INTEGER
        FORMAT VARIABLE FVT
        PROGRAM COMMON LZERO,RZERO
        D'N RES(50),AP(4000,APDIM) ,TPROB(4500),LOC1(17),LIST(200)
        V'S APDIM=2,1,0
        EQUIVALENCE (APDIM(2),NEL102)
        NEL102=(NEL+1)/2
R
R    CONVERT  SSM
R
        L=1
        CTT1=T3(1)
        REWIND TAPE 10
        REWIND TAPE 3
        AT=0
        T'H  LOP1,FOR B=1,1,B.G.NS
        P'S B
        ORD=0
        C1=T3(B)
        C2=C1
        C3=C1
        T'H LOP2,FOR C1=C1,1,T1(C1).E.0
        T2(C1)=1.LS.T2(C1)
LOP2    ORD=ORD+1
```

Figure B-28. RAP03 Subprogram, Program Listing (1 of 6)

```
         P'S  ORD
         W'R L.G.4000/((NEL+1)/2)
FIVE5    REWIND TAPE 10
         REWIND TAPE 3
         W'R TAPENO.E.2
         SETEOF.(RTP2)
RTP2     READ BINARY TAPE 10,AP(1,1)...AP(4000/((NEL+1)/2),(NEL+1)/2)
         O'E
         SETEOF.(RTP3)
RTP3     READ BINARY TAPE 3,AP(1,1)...AP(4000/((NEL+1)/2),(NEL+1)/2)
         E'L
         T'O SIX6
         E'L
         T'H LOPOUT,FOR C=1,1,C.G.L
         T'H LOPIN,FOR CC=1,1,CC.G.(NEL+1)/2
LOPIN    AP(C,CC)=A(C,CC)
LOPOUT   CONTINUE
        R
        R STRIP ROUTINE
        R
         PRINT COMMENT $0   START   STRIP$
         OVER=0
SIX6     Q=0
         X=0
         SAVC2=C2
         T'H LOP3,FOR M=1,1,M.G.L
         P'S  M
         W'R M.G.4000/((NEL+1)/2)
         M=1
         L=L-4000/((NEL+1)/2)
         W'R TAPENO.E.2
         SETEOF.(RDT2)
RDT2     READ BINARY TAPE 10,AP(1,1)...AP(4000/((NEL+1)/2),(NEL+1)/2)
         O'E
         SETEOF.(RDT3)
RDT3     READ BINARY TAPE 3,AP(1,1)...AP(4000/((NEL+1)/2),(NEL+1)/2)
         E'L
         E'L
         NOS=0
         K=0
INCK     K=K+1
         W'R (T1(C2)/2)*2.E.T1(C2)
         WA=(AP(M,T1(C2)/2).LS.18).RS.18
         O'E
         WA=AP(M,(T1(C2)+1)/2).RS.18
         E'L
         RES(K)=WA.EV.T2(C2)
         W'R RES(K).G.WA
         NOS=1
         T'O GO
         E'L
         W'R K.E.ORD
         T'O GO
         E'L
         C2=C2+1
         T'O INCK
GO       C2=SAVC2
         SAVC3=C3
         T'H LOP5,FOR P=1,1,P.G.K
         P'S  P
```

Figure B-28. RAP03 Subprogram, Program Listing (2 of 6)

```
            W'R RES(P).E.0.TRANSFER TO LOP5
            Q=Q+1
            W'R Q.G.4000/((NEL+1)/2)
            W'R TAPENO.E.2
            WRITE BINARY TAPE 3      .A(1.1)...A(Q.(NEL+1)/2)
            O'E
            WRITE BINARY TAPE 10     .A(1.1)...A(Q.(NEL+1)/2)
            E'L
            AT=AT+1
            Q=1
            E'L
            W'R P.E.K.TRANSFER TO TESTS1
BACK        T'H LOP6.FOR C=1.1.C.G.(NEL+1)/2
LOP6        A(Q.C)=AP(M.C)
            W'R (T1(C3)/2)*2.E.T1(C3)
            A(Q.T1(C3)/2)=RES(P).V.(A(Q.T1(C3)/2).A.RZERO)
            AP(M.T1(C3)/2)=T2(C3).V.(AP(M.T1(C3)/2).A.RZERO)
            O'E
            A(Q.(T1(C3)+1)/2)=(RES(P).LS.18).V.(A(Q.(T1(C3)+1)/2).A.
           1LZERO)
            AP(M.(T1(C3)+1)/2)=(T2(C3).LS.18).V.(AP(M.(T1(C3)+1)/2).A.
           1LZERO)
            E'L
LOP5        C3=C3+1
            X=X+1
            P'S  X
            T'H LOP7.FORC=1.1.C.G.(NEL+1)/2
LOP7        SORT(X.C)=AP(M.C)
            T'O LOP3
TESTS1      W'R NOS.E.1
            T'H LOP8.FORC=1.1.C.G.(NEL+1)/2
LOP8        A(Q.C)=AP(M.C)
            T'O LOP3
            O'E
            T'O BACK
            E'L
LOP3        C3=SAVC3
            W'R AT.LE.0
            L=Q
            T'O FOUR4
            O'E
            L=Q+AT*(4000/((NEL+1)/2))
            W'R TAPENO.E.2
            WRITE BINARY TAPE 3      .A(1.1)...A(Q.(NEL+1)/2)
            TAPENO=3
            END OF FILE TAPE 3
            O'E
            WRITE BINARY TAPE 10     .A(1.1)...A(Q.(NEL+1)/2)
            TAPENO=2
            END OF FILE TAPE 10
            E'L
            AT=0
            E'L
FOUR4       CONTINUE
           R              EQUATION GENERATOR              NOT A SUBROUTINE
           R
            TPROB=0
            SET LIST TO TPROB
            SAVE DATA $T$.$P$.$($
            INSERT.(B)
```

Figure B-28. RAP03 Subprogram, Program Listing (3 of 6)

```
          SAVE DATA $)$,$=$
          SAVE DATA $B$,$P$,$($
          INSERT.(B)
          SAVE DATA $)$,$*$,$($
          W'R X.E.0
          SAVE DATA $0$,$.$
          CTT1=T3(B+1)
          T'O LABEL
          E'L
          T'H LOPP1,FOR J=1,1,J.G.X
          W'R J.G.1
          SAVE    DATA $+$
          E'L
          REM=CTT1
          T'H LOPP2,FOR K=1,1,K.G.(NEL+1)/2
         R   CHECK FOR OVERFLOW OF TPROB TABLE
          W'R TPROB+90.G.4500
          OVER=OVER+1
LL80      T'H LL80,FOR JJJ=1,1,TPROB(JJJ).E.$+$.AND.TPROB(JJJ+3).NE.TPR
         1OB(JJJ-4)
          TPROB(JJJ)=$)$
          SAVJ=JJJ
          KKK=0
          T'H LL81,FOR JJJ=JJJ+1,1,JJJ.G.TPROB
          KKK=KKK+1
LL81      LIST(KKK)=TPROB(JJJ)
          TPROB=SAVJ
          EXECUTE EQLOAD.(TPROB)
LL82      T'H LL82,FOR JJJ=1,1,TPROB(JJJ).E.$=$
          FVT=JJJ+4
          WRITE BCD TAPE9,FMT8,$X$,$($,OVER,$)$,$=$,TPROB(1)...TPROB(JJ
         1J-1)
          V'S FMT8=$S11,'FVT'C1,T80*$
          TPROB=2*JJJ+1
          SET LIST TO TPROB
          T'H LL83,FOR JJJ=1,1,JJJ.G.KKK
LL83      SAVE DATA LIST(JJJ)
          E'L
          WRK=SORT(J,K).A.RZERO
          W'R WRK.E.AVAIL(K).A.RZERO
TWO       WRK=SORT(J,K).A.LZERO
          W'R WRK .E. AVAIL(K).A.LZERO,TRANSFER TO LOPP2
          KK=1
          W'R T1(CTT1).E.2*K
          WRK=WRK.EV.T2(CTT1)
          CTT1=CTT1+1
          W'R WRK.E.0,T'O LOPP2
          T'O ONE
          E'L
          O'E
          WRK=WRK.RS.18
          KK=0
          W'R T1(CTT1).E.2*K-1
          WRK=WRK.EV.T2(CTT1)
          CTT1=CTT1+1
          W'R WRK.E.0,T'O TWO
          T'O ONE
          E'L
          E'L
ONE       SHIFT=0
```

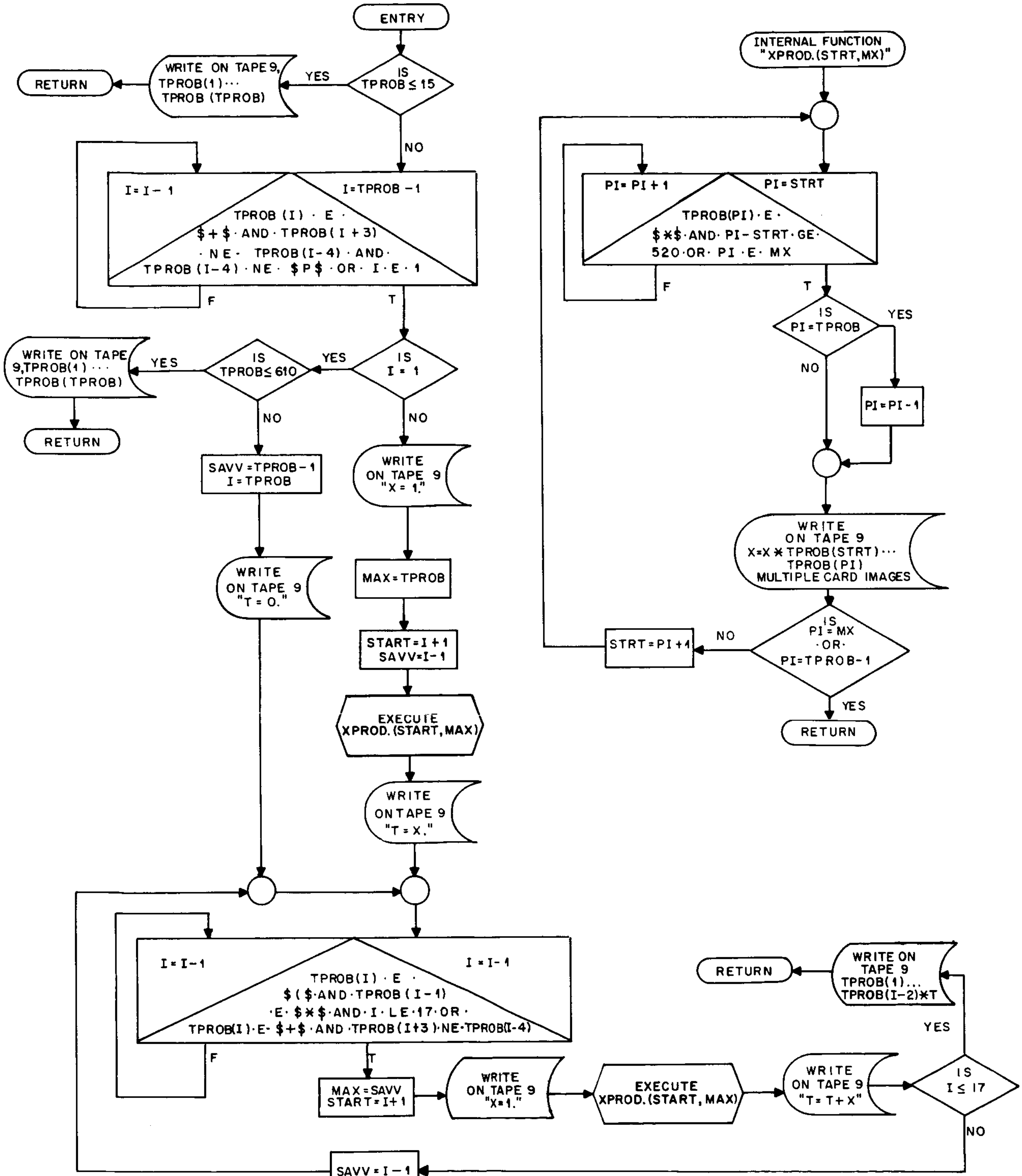
Figure B-28. RAP03 Subprogram, Program Listing (4 of 6)

```
        COUNT=0
        EXECUTE ZERO.(LOC1(0),...,LOC1(17))
FOUR    W'R WRK.A.1.E.1
        LOC1(COUNT)=SHIFT
        COUNT=COUNT+1
        E'L
        WRK=WRK.RS.1
        SHIFT=SHIFT+1
        W'R SHIFT.G.SDA(2*K-1+KK),TRANSFER TO THREE
        T'O FOUR
THREE   W'R COUNT*2.G. SDA(2*K-1+KK)
        W'R COUNT.E.1,TRANSFER TO PRDUCT
        T'O ONEMIN
        E'L
        W'R COUNT.G.1
        SAVE    DATA $($
        E'L
PRDUCT  T'H LOPP3,FOR LL=0,1,LL.GE.COUNT
        W'R LOC1(LL).E.0
        SAVE DATA $P$,$P$,$($
        INSERT.(2*K-1+KK)
        SAVE DATA $)$
        O'E
        SAVE DATA $P$,$($
        INSERT.(2*K-1+KK)
        SAVE DATA $,$
        INSERT.(LOC1(LL))
        SAVE DATA $)$
        E'L
        W'R LL+1.E.COUNT,TRANSFER TO LOPP3
        SAVE    DATA $+$
LOPP3   CONTINUE
        W'R COUNT.G.1
        SAVE    DATA $)$
        E'L
SEV     W'R K.G.(NEL+1)/2,TRANSFER TO SIX
        SAVE    DATA $*$
SIX     W'R KK.E.0
        W'R K.G.(NEL+1)/2,TRANSFER TO LOPP2
        T'O TWO
        E'L
        T'O LOPP2
ONEMIN  SAVE DATA $($,$1$,$,$,$-$
        W'R SDA(2*K-1+KK)+1-COUNT.G.1
        SAVE    DATA $($
        E'L
        MM=0
        T'H LOPP4,FOR LL=0,1,MM.G.SDA(2*K-1+KK)
        W'R LOC1(LL).E.MM,TRANSFER TO LOPP4
        W'R MM.E.0
        SAVE DATA $P$,$P$,$($
        INSERT.(2*K-1+KK)
        SAVE DATA $)$
        O'E
        SAVE DATA $P$,$($
        INSERT.(2*K-1+KK)
        SAVE DATA $,$
        INSERT.(MM)
        SAVE DATA $)$
        E'L
```

Figure B-28. RAP03 Subprogram, Program Listing (5 of 6)

```
          W'R MM.E. SDA(2*K-1+KK),TRANSFER TO LOPP4
          LL=LL-1
          SAVE     DATA $+$
LOPP4     MM=MM+1
          W'R TPROB(TPROB).E.$+$,TPROB=TPROB-1
          W'R SDA(2*K-1+KK)+1-COUNT.G.1
          SAVE     DATA $)$
          E'L
          SAVE     DATA $)$
          T'O SEV
LOPP2     CONTINUE
          W'R J.L.X.CTT1=REM
          W'R TPROB(TPROB)   .E. $*$,TPROB=TPROB-1
LOPP1     CONTINUE
          CTT1=T3(B+1)
LABEL     W'R TPROB(TPROB).E.$($
          TPROB=TPROB-2
          O'E
          SAVE DATA $)$
          E'L
R        END OF EQUATION GENERATOR
R      TEST FOR WHETHER OVERFLOW ROUTINE WAS NEEDED EARLIER
          W'R OVER.E.0
          EXECUTE EQLOAD.(TPROB)
          T'O ZETA
          E'L
LL90      T'H LL90,FOR JJJ=1,1,TPROB(JJJ).E.$=$
          FVT=2*JJJ+4
PLACE     WRITE BCD TAPE9,FMT8,TPROB(1)...TPROB(JJJ),TPROB(1)...TPROB(J
1JJ-1),$+$,$X$,$($,OVER,$)$
          OVER=OVER-1
          W'R OVER.E.0,T'O ZETA
          T'O PLACE
ZETA      PRINT BCD RESULTS TPROB(1)...TPROB(TPROB)
          W'R L.E.0
          T'H FINI,FOR B=B+1,1,B.G.NS
          TPROB=0
          SET LIST TO TPROB
          SAVE DATA $T$,$P$,$($
          INSERT.(B)
          SAVE DATA $)$,$=$,$0$,$.$
FINI      EXECUTE EQLOAD.(TPROB)
          T'O ENDRP3
          E'L
LOP1      PRINT OCTAL RESULTS A(1,1)...A(L,(NEL+1)/2),
1SORT(1,1)...SORT(X,(NEL+1)/2)
          INTERNAL FUNCTION (QB)
          ENTRY TO INSERT.
          QQ=QB/10
          W'R QQ.L.1,T'O QQON
          SAVE DATA  N(QQ)
QQON      QQ=QB-10*QQ
          SAVE DATA N(QQ)
          V'S N=$0$,$1$,$2$,$3$,$4$,$5$,$6$,$7$,$8$,$9$
          F'N
          E'N
ENDRP3    F'N
          E'N
```

Figure B-28. RAP03 Subprogram, Program Listing (6 of 6)



ARGUMENTS	
INPUT	OUTPUT
TPROB	TP EQ'N VIA TAPE 9

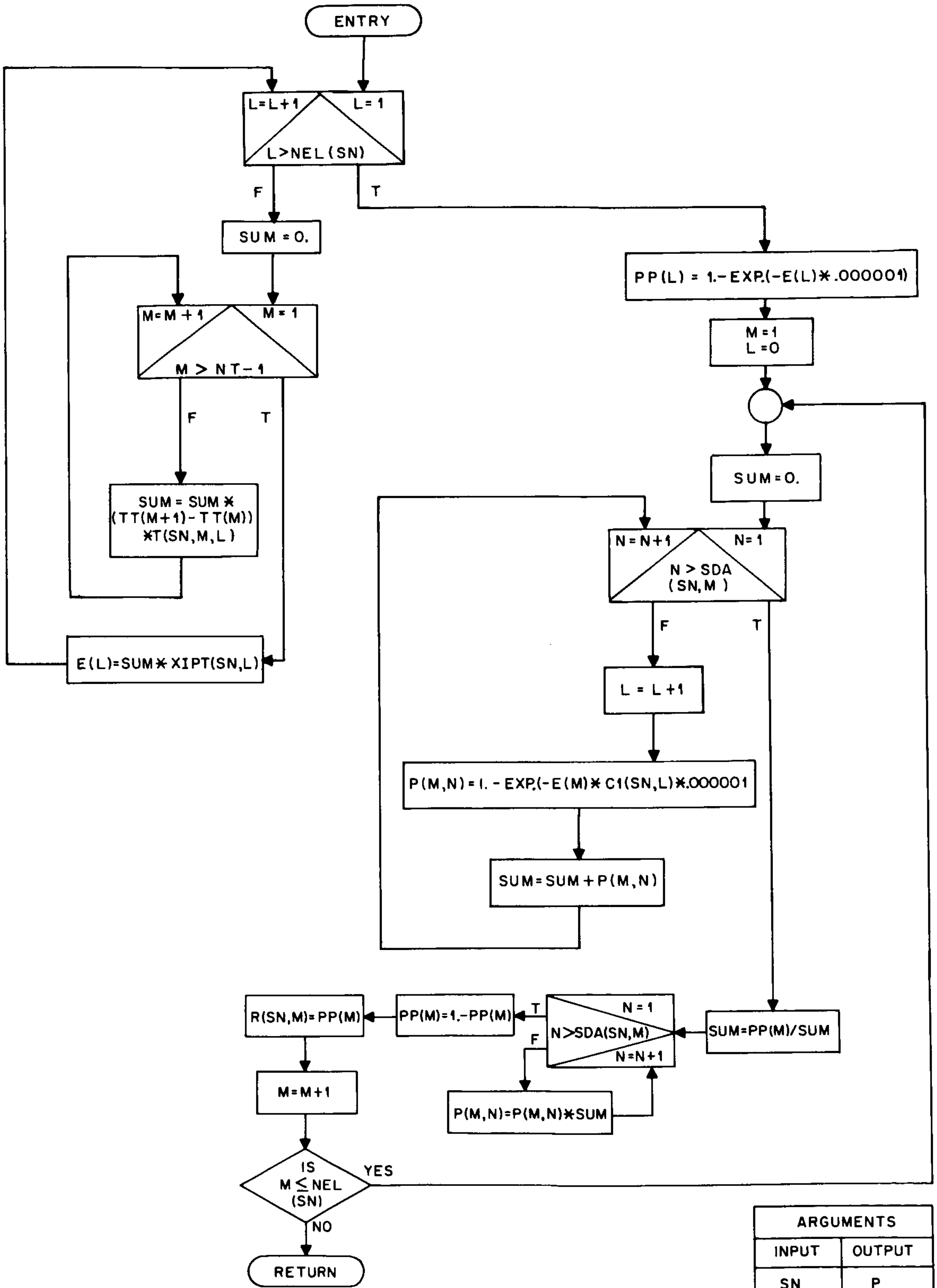
Figure B-29. EQLOAD Subprogram, Logic Flow Chart

```
R          EQUATION LOADER -EQLOAD.
R
R    ....DEFINITION OF PROGRAM VARIABLES....
R
R CMARK                     ARRAY OF PRESET BCD CHARACTERS
R MAX                       MAXIMUM POSITION OF STRING IN TPROB TO BE
R                           LOADED IN CARD IMAGE FORM
R SAVV                      LAST POSITION TO BE SAVED IN TPROB BEFORE
R                           STRING DEFINED BY (START,MAX)
R START                     LOWEST MAXIMUM POSITION OF STRING IN TPROB TO
R                           BE LOADED IN CARD IMAGE FORM
R
R    ....PROGRAM LISTING....
R
           EXTERNAL FUNCTION(TPROB)
           ENTRY TO EQLOAD.
           FORMAT VARIABLE FOMVAR
           N'S INTEGER
           D'N CMARK(10)
           V'S CMARK=$ $,$1$,$2$,$3$,$4$,$5$,$6$,$7$,$8$,$9$
           W'R TPROB.LE. 15
           FOMVAR=TPROB+1
           WRITE BCD TAPE9,FMT1,CMARK,TPROB(1)...TPROB(TPROB)
           V'S FMT1=$S10,'FOMVAR'C1,T80*$
           T'O ENDEQL
           E'L
L1         T'H L1,FOR I=TPROB-1,-1, TPROB(I).E.$+$.AND.TPROB(I+3).NE.TPR
          1OB(I-4).AND.TPROB(I-4).NE.$P$.OR. I.E.1
           W'R I.E.1
           W'R TPROB.LE.610
           KJ=0
           II=0
           T'H L3,FOR OT=1,-OT+II+1,II.GE.TPROB
           II=II+61
           W'R II.G.TPROB,II=TPROB
           FOMVAR=II-OT+2
           WRITE BCD TAPE9,FMT1,CMARK(KJ),TPROB(OT)...TPROB(II)
L3         KJ=KJ+1
           T'O ENDEQL
           E'L
           SAVV=TPROB-1
           I=TPROB
           FOMVAR=5
           WRITE BCD TAPE9,FMT1,CMARK,$T$,$=$,$0$,$.$
           T'O L2
           E'L
           FOMVAR=5
           WRITE BCD TAPE9,FMT1,CMARK,$X$,$=$,$1$,$.$
           MAX=TPROB
           START=I+1
           SAVV= I - 1
           XPROD.(START,MAX)
           FOMVAR=4
           WRITE BCD TAPE9,FMT1,CMARK,$T$,$=$,$X$
```

Figure B-30. EQLOAD Subprogram, Program Listing (1 of 2)

```
L2        T'H L2,FOR I=I-1,-1,TPROB(I).E.$($.AND.TPROB(I-1).E.$*$.AND.I
         1.LE.17.OR.TPROB(I).E.$+$.AND.TPROB(I+3).NE.TPROB(I-4)
          MAX=SAVV
          START=I+1
          FOMVAR = 5
          WRITE BCD TAPE 9,FMT1,CMARK,$X$,$=$,$1$,$.$
          XPROD.(START,MAX)
          FOMVAR=6
          WRITE BCD TAPE9,FMT1,CMARK,$T$,$=$,$T$,$+$,$X$
          W'R I.LE.17
          FOMVAR=I+1
          WRITE BCD TAPE9,FMT1,CMARK,TPROB(1)...TPROB(I-2),$*$,$T$
          T'O ENDEQL
          E'L
          SAVV=I-1
          T'O L2
          INTERNAL FUNCTION (STRT,MX)
          ENTRY TO XPROD.
LL1       T'H LL1,FOR PI=STRT ,1,TPROB(PI).E.$*$.AND.PI-STRT.GE.520
         1.OR.PI.E.MX
          W'R PI.E.TPROB,PI=PI-1
          II=56
          W'R II.G.PI-STRT,II=PI-STRT
          FOMVAR=II+6
          WRITE BCD TAPE9,FMT1,CMARK,$X$,$=$,$X$,$*$,TPROB(STRT )...TPR
         1OB(STRT+II)
          W'R II.L.PI
          KJ=1
          T'H LL2,FOR OT=STRT+II+1,II+1,II.GE.PI-OT+II+1
          II=60
          W'R II.G.PI-OT,II=PI-OT
          FOMVAR=II+2
          WRITE BCD TAPE9,FMT1,CMARK(KJ),TPROB(OT)...TPROB(OT+II)
LL2       KJ=KJ+1
          E'L
          W'R PI.E.MX .OR.PI.E.TPROB-1
          FUNCTION RETURN
          E'L
          STRT =PI+1
          T'O LL1
          E'N
ENDEQL    F'N
          E'N
```

Figure B-30. EQLOAD Subprogram, Program Listing (2 of 2)



ARGUMENTS	
INPUT	OUTPUT
SN	P PP

Figure B-31. PEXP Subprogram, Logic Flow Chart

```
R             CALCULATION OF P'S USING EXPONENTIAL PEXP.
R
R    ....DEFINITION OF PROGRAM VARIABLES....
R
R C1(10*15)                    SAME AS CORE1
R E(30)                        CALCULATED EXPONENT
R L                            CURRENT LOCATION IN C1 ARRAY FOR CURRENT SYS.
R M                            CURRENT ELEMENT NO.
R N                            CURRENT MODE NO. FOR ELEMENT M
R NEL(10)                      SAME AS CORE1
R NS(10)                       SAME AS CORE1
R NT                           SAME AS CORE1
R SDA(10*15)                   SAME AS CORE1
R SUM                          WORKING VARIABLE FOR SUMMING K FACTORS - ALSO
R                              FOR SUMMING PP'S
R T(10*10*15)                  SAME AS CORE1
R TT(10*15)                    SAME AS CORE1
R XL(15)                       SAME AS CORE1
R
R    ....PROGRAM LISTING....
R
       EXTERNAL FUNCTION(P,PP,SN)
       E'O PEXP.
       I'R L,M,N,NT,SDA,NEL,SN,J,SET.
       V'S XDIM=SET.,0,0
       P'N NEL(10),NS(10),NT,TT(10),T(10*10*15),XIPT(150,XDIM),
      1C1(10*15),SDA(10*15),R(10*15),LD(46),LS(8),MO,DAY,YR,REL,NAME
      2(750),A(10*15),B(10*15),C(10*15)
       D'N E(30)
       T'H LA,FOR L=1,1,L.G.NEL(SN)
       SUM=0.
       T'H LB,FOR M=1,1,M.G.NT-1
LB     SUM=SUM+(TT(M+1)-TT(M))*T(SN,M,L)
       E(L)=SUM*XIPT(SN,L)
LA     PP(L)=1.-EXP.(-E(L)*.000001)
       M=1
       L=0
AGAIN  SUM=0.
       T'H LC,FOR N=1,1,N.G.SDA(SN,M)
       L=L+1
       P(M,N)=1.-EXP.(-E(M)*C1(SN,L)*.000001)
LC     SUM=SUM+P(M,N)
       SUM=PP(M)/SUM
       T'H LD,FOR N=1,1,N.G.SDA(SN,M)
LD     P(M,N)=P(M,N)*SUM
       PP(M)=1.-PP(M)
       R(SN,M)=PP(M)
       M=M+1
       W'R M.LE.NEL(SN),T'O AGAIN
       F'N
       E'N
```

Figure B-32. PEXP Subprogram, Program Listing

3.3 IBM 1620 VERSION

This section describes the methodology utilized for system reliability simulation as programmed for the IBM 1620*. Each step is illustrated relative to a hypothetical system. Two computer source languages have been used in programming the Reliability Simulator. FORTRAN II-D has been used wherever possible, while SPS-II-D has been used where memory capacity or other limitations require a lower level language.

3.3.1 Introduction

Reliability Simulation of large systems is achieved on the IBM 1620 by repeated application of the RAPID technique to simulate and analyze subdivisions of the system. The Reliability Simulator is comprised of two main programs which employ numerous subprograms to perform the simulation and analysis, respectively. One program (RAPID1) controls the simulation phase and the other program (RAPID2) controls the analysis phase of RAPID. These two programs are executed alternately until all input data have been processed and the system analysis is complete.

---

*The IBM 1620 system, referred to in this report as the IBM 1620, consists of the following:

1. 1620-II Central Processing Unit with Index Registers

1 1625-3 Magnetic Core Memory

1 1622 Card Reader-Punch

1 1443 Printer

3 1311 Disk Drives

This configuration constitutes the minimum IBM 1620 system on which this version of the RAPID technique can be executed.

3.3.2 System Description

Consider the illustration of the system in Figure B-33. SYSTEM is composed of three elements E1, SUBS2, and SUBS3. SUBS2 is composed of two elements, E1 and E2, while SUBS3 is made up of the two elements, SUBS1 and E2. SUBS1 is itself made up of two elements, E1 and E2.

SUBS1, SUBS2, and SUBS3 are so designated to indicate that they are to be considered as "subsystems," which will be individually simulated and analyzed. A "building block" approach to reliability simulation of SYSTEM will be applied, in which each subsystem is treated as an element after the mode probabilities have been computed from the simulation and analysis of that subsystem.

The reliability for SUBS1 is first determined in terms of element mode probabilities for its elements E1 and E2. SUBS1 is now referred to as an element of SUBS3 along with the E2 associated with SUBS3. The probabilities previously calculated for SUBS1 are used along with the element mode probabilities for E2 to determine the probability of failure and hence the reliability of SUBS3. SUBS3 can then be treated as an element of the system SYSTEM.

For example, the probabilities of occurrence of the modes established by analyzing SUBS1 are used as element mode probabilities in evaluating the system mode models of the subsystem SUBS3, since SUBS1 is an element of SUBS3 anong with E2. This procedure is repeated for SUBS3, leading to the final analysis of SYSTEM.

3.3.3 System Inputs

The RAPID Analysis Data Transmittal Forms are used for inputting data for the reliability simulation. Inputs for the subsystems analyses are placed one behind the other. If the system has only one level of element (no elements below the subsystem level), the input is comprised of only one "system" to be simulated. If an error is detected in a set of data for an analysis, that analysis is suppressed and the remaining cards of data in that set are ignored.

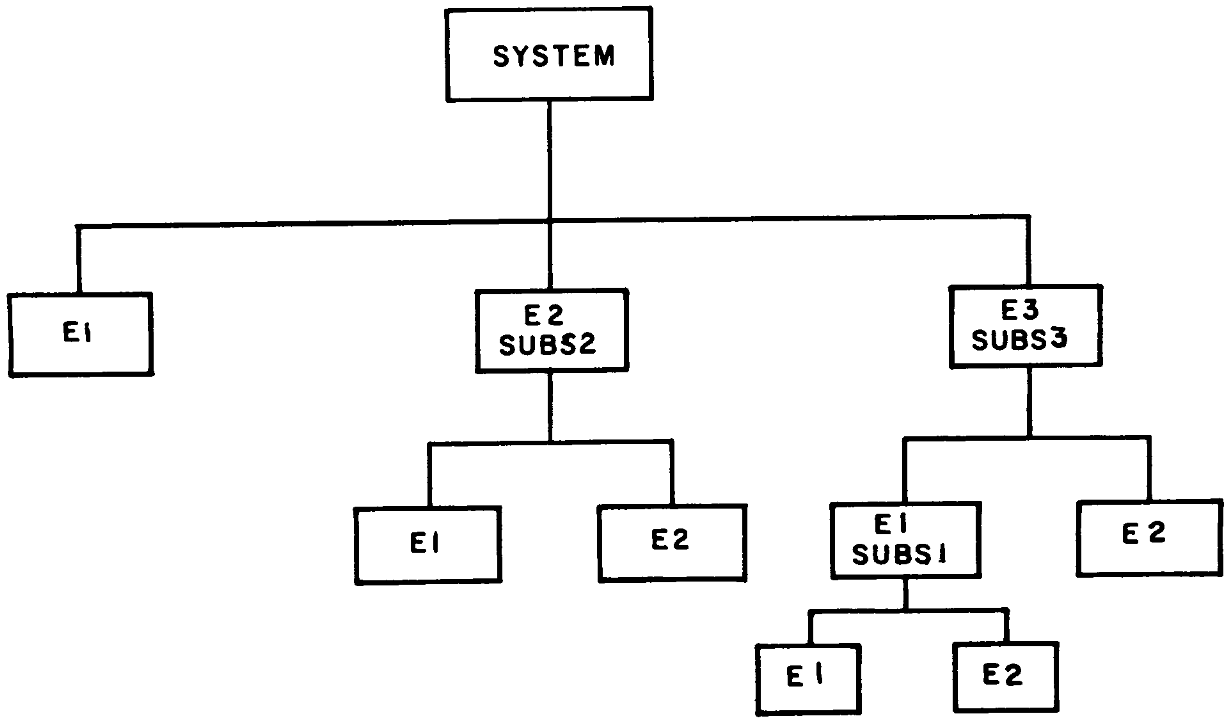


Figure B-33. Example System for IBM 1630 Reliability Simulator Discussion

For the reliability simulation and analysis of a multilevel system such as that shown in Figure B-33, the order in which each system is analyzed is important. In the hypothetical system, assume that SUBS1 and SUBS2 have two system failure modes and that SUBS3 has only one failure mode.

The order of analysis to analyze the system SYSTEM would be as follows;

1	SUBS1	SYSTEM MODE 1
2	SUBS1	SYSTEM MODE 2
3	SUBS1	SYSTEM MODE 0
4	SUBS3	SYSTEM MODE 1
5	SUBS3	SYSTEM MODE 0
6	SUBS2	SYSTEM MODE 1
7	SUBS2	SYSTEM MODE 2
8	SUBS2	SYSTEM MODE 0
9	SYSTEM	SYSTEM MODE 1

For each simulation and analysis, the element mode probabilities must be defined. That is, each element must have exponential failure rate data or actual probabilities on cards supplied, or it must have been previously analyzed as a "system," including an analysis of a non-failure mode.

All analysis and calculation of probabilities associated with a given system, whether at a lower element level, subsystem level, or at the system level, must be made for the same points in time. That is, the type 4 cards, shown on Data Transmittal Form I, should be identical for each individual simulation and analysis associated with a given system.

Sample Data Transmittal Forms are shown in Figures B-34 through B-41 as typical inputs for the subsystems SUBS1 and SUBS3 to illustrate the use of the S code. The elements E1 and E2 of the subsystem SUBS1 have their element mode probabilities defined with exponential failure rate data (S = 1). The element E2 of the subsystem SUBS3 has its element mode probabilities read in from cards (S = 2). Note that the first element of SUBS3 has an S code of zero, indicating that this element (SUBS1) has been previously analyzed, and the mode probabilities are available in disk storage.

3.3.4 System Simulation

Before a new system reliability simulation and analysis is started, the table of stored probabilities, the summary table, and the

LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF I	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET 1 of 4

TITLE CARD 1

CODE (cols 4–11)	SYSTEM NAME (cols 18–63)	DATE (cols 70–77)	Col 80
SUBS1	SUBSYSTEM 1 - ELEMENT 1 OF SUBSYSTEM 3	9 1 65	1

TITLE CARD 2

H (col 3)	MODE NAME (cols 18–63)	Col 80
2	SECOND FAILURE MODE	2

SYSTEM DEFINITION CARD

NE (cols 1–3)	ELEMENT NUMBER (cols 21–70)	Col 80
2	24	3

TIME CARDS

NT	TIME	J	TIME	J	TIME	J	TIME	J	TIME	J	Col 80
3	0.0	1	10.0	1	30.0						4
											4
											4
											4
											4

Figure B-34. Sample Data Transmittal Form (DTF I) for SUBS1 Subsystem

LSI	LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF II	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET 2 of 4

SYSTEM MODE ARRAY CARDS

NS (cols 1–3)	Element Number 1 (col 21)	Element Number 2 (col 22)	Col 80
4	1		5
		1	5
		2	5
	2	4	5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5

Figure B-35. Sample Data Transmittal Form (DTF II) for SUBS1 Subsystem

LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF III	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET 3 of 4

ELEMENT CARDS

EN	NAME	S	LAMBDA	CODE	
1	ELEMENT A	1	10.4	ELA	6

	MODE NAME	C		MODE NAME	C		MODE NAME	C	
1	MODE A 1	.3	2	MODE A 2	.7	3			6
4			5			6			6
7			8			9			6

EN	NAME	S	LAMBDA	CODE	
2	ELEMENT B	1	27.41	ELB	6

	MODE NAME	C		MODE NAME	C		MODE NAME	C	
1	MODE B 1	.4	2	MODE B 2	.2	3	MODE B 3	.2	6
4	MODE B 4		5			6			6
7			8			9			6

EN	NAME	S	LAMBDA	CODE	
					6

	MODE NAME	C		MODE NAME	C		MODE NAME	C	
1			2			3			6
4			5			6			6
7			8			9			6

Figure B-36. Sample Data Transmittal Form (DTF III) for SUBS1 Subsystem

LSI	LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF IIIA	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET 4 of 4

ELEMENT K FACTOR CARDS

EB (2–3)	EE (6–7)	K FACTOR (10–19)	K FACTOR (24–33)	K FACTOR (38–47)	K FACTOR (52–61)	K FACTOR (66–75)	80
1	2	10.0	1.5				7
							7
							7
							7
							7

EB (2–3)	EE (6–7)	K FACTOR (10–19)	K FACTOR (24–33)	K FACTOR (38–47)	K FACTOR (52–61)	K FACTOR (66–75)	80
							7
							7
							7
							7
							7

EB (2–3)	EE (6–7)	K FACTOR (10–19)	K FACTOR (24–33)	K FACTOR (38–47)	K FACTOR (52–61)	K FACTOR (66–75)	80
							7
							7
							7
							7
							7

Figure B-37. Sample Data Transmittal Form (DTF IIIA) for SUBS1 Subsystem

LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF I	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET 1 of 4

TITLE CARD I

CODE (cols 4–11)	SYSTEM NAME (cols 18–63)	DATE (cols 71–77)	Col 80
SUBS3	SUBSYSTEM 3 - ELEMENT 3 OF LARGE SYSTEM	9 1 65	1

TITLE CARD 2

H (col 3)	MODE NAME (cols 18–63)	Col 80
1	FAILURE MODE	2

SYSTEM DEFINITION CARD

NE (cols 1–3)	ELEMENT NUMBER (cols 21–70)	Col 80
2	21	3

TIME CARDS

NT	TIME	J	TIME	J	TIME	J	TIME	J	TIME	J	Col 80
3	0.0	1	10.0	1	30.0						4
											4
											4
											4
											4

Figure B-38. Sample Data Transmittal Form (DTF I) for SUBS3 Subsystem

SYSTEM MODE ARRAY CARDS

NS (cols 1–3)	Element Number 1 (col 21)	Element Number 2 (col 22)	Col 80
2	1		5
	2	1	5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5
			5

Figure B-39. Sample Data Transmittal Form (DTF II) for SUBS3 Subsystem

LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF III	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET 3 of 4

ELEMENT CARDS

EN	NAME	S	LAMBDA	CODE	80
1	SUBSYSTEM	0		SUBS1	6

No.	MODE NAME	C	No.	MODE NAME	C	No.	MODE NAME	C	80
1	FAILURE MODE 1		2	FAILURE MODE 2		3			6
4			5			6			6
7			8			9			6

EN	NAME	S	LAMBDA	CODE	80
2	ELEMENT C	2		ELC	6

No.	MODE NAME	C	No.	MODE NAME	C	No.	MODE NAME	C	80
1	MODE C 1		2			3			6
4			5			6			6
7			8			9			6

EN	NAME	S	LAMBDA	CODE	80
					6

No.	MODE NAME	C	No.	MODE NAME	C	No.	MODE NAME	C	80
1			2			3			6
4			5			6			6
7			8			9			6

Figure B-40. Sample Data Transmittal Form (DTF III) for SUBS3 Subsystem

LEAR SIEGLER, INC. INSTRUMENT DIVISION	RAPID DTF IIIB	PROBABILISTIC SYSTEM MODE ANALYSIS	SHEET 4 of 4

ELEMENT MODE PROBABILITY CARDS

EN	M	TIME	PROBABILITY	TIME	PROBABILITY	TIME	PROBABILITY	80
2	1	0.0	0.0	5.0	0.01	10.0	0.02	8
2	1	20.0	0.06	30.0	0.11			8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8
								8

Figure B-41. Sample Data Transmittal Form (DTF IIIB) for SUBS3 Subsystem size of the combined System Mode Array must be initialized. This is accomplished by executing the routine RAP022. If the new analysis requires information previously stored on the disk files, RAP022 should not be executed.

The simulation phase begins with the reading of the type 1 through 5 input cards for mode 1 of the subsystem SUBS1. If no errors are detected in these cards, the model for the probability of occurrence of this system mode is generated and stored in a work area on the disk files. The remaining inputs for the analysis of SUBS1 -- system mode 1 are then read in and checked for errors.

If no errors are detected in the remaining cards of this data set, the subroutine RAP016 is constructed which will contain the mathematical model for the mode under consideration. This is constructed in FORTRAN II-D source language and stored in card image format in a work area on disk files. The following statements are the first to be generated by the simulator:

```
*LD1SKRAP016

      SUBROUTINE RAP016 (BP,TR,P)

      DIMENSION BP (100), TR(100), P(50,10)
```

The first statement above is a control statement indicating to the FORTRAN II-D Compiler that a relocatable object program is to be stored on the disk files and identified by the name RAP016, following a successful compilation. The second statement is the FORTRAN statement indicating this program is a subroutine with the three variable arguments BP, TR, and P. The third statement is the FORTRAN statement indicating the three arguments are arrays and gives the size of these arrays.

The next series of statements in the subroutine comprise the mathematical model for mode 1 of SUBS1. The first statements in this series are arithmetic statements which will calculate the Basic Submode Probabilities of the system. These are designated as BP in the subroutine. Following the BP equations are arithmetic statements which will calculate the Total Row Probabilities of the system. These probabilities are referred to as the array TR.

Both the Basic Submode Probabilities and Total Row Probabilities are expressed in terms of element mode probabilities. These are represented by P(I, J) where I is the element number and J represents the element mode number. It should be noted that J is always equal to the subject element mode number plus one. This is necessitated by the fact that element mode zero cannot be represented by a subscript equal to zero because zero subscripts are not permitted in the FORTRAN II-D language.

After the mathematical model has been inserted, the subroutine is completed with two concluding statements:

```
RETURN

END    .
```

These statements will effect a branch back to the calling routine from subroutine RAP016 and signal the FORTRAN compiler that the end of the source statements has been encountered.

This system mode is then detected to be the first in possibly a series of modes for the system SUBS1. To effect simulation of a non-failure system mode, the current System Mode Array is stored in a communications region on the disk files. The system code SUBS1 is then saved for comparison with succeeding system codes.

With the simulation phase completed for mode 1 of SUBS1, the first outputting is started. The System Definition Array and the System Mode Array are printed as the first page of the output. Values of variables and arrays which will be needed in the analysis phase of RAPID are then stored in a communication region on the disk files.

A call to the 1620 Monitor system is given by a CALL MONITR statement. This will result in the subroutine RAP016 to be compiled and stored in relocatable object form on the disk files. At this point, RAPID2 is loaded into core memory and its execution begins.

The analysis of the current system mode is then carried out by evaluating the model at the points in time specified in the type 4 cards. The element mode probabilities, in this case, are calculated from an exponential failure rate distribution, since the S code is 1 for all elements of SUBS1. The results of the analysis are printed and the probabilities of occurrence and non-occurrence for mode 1 of SUBS1 are then stored in tabular form on the disk files. Upon completion of this phase, the first routine (RAPID1) is reloaded into core memory for the simulation phase on the next data set.

Mode 2 of SUBS1 is processed in the same manner as the previous mode. The mode 2 SMA is combined with the mode 1 SMA for later use in processing mode 0 for SUBS1. RAPID1 is then loaded back into core memory and execution begun.

With this execution of the simulation phase, the type 1 through 4 cards for mode 0 of the subsystem SUBS1 are read. The system mode 0 (non-failure mode) is handled in a slightly different manner than the non-zero system modes. For a system mode of zero, the type 5 cards (SMA) are omitted. The model and subroutine generation is performed on the combined System Mode Array comprised of the SMA's from mode 1 and mode 2, which are stored in the communication region on the disk files. The printing of the System Mode Array is omitted, since this combined SMA actually is the one for the combined modes 1 and 2, and not for mode 0. The computer programs STRIP this combined SMA and generate the mathematical model for mode 0 from the results of the logical intersection of all possible system states with this SMA. That is, the remainder from the intersection is the set of system states representing mode 0.

RAPID2 is then executed for mode 0 in the same manner as in the previous cases. There is no printed analysis for each point in time as is the case with a non-zero system modes. However, a summary table and element descriptions are printed as usual, and the mode 0 probabilities of occurrence and non-occurrence are stored in tabular form on the disk files.

At this point, the subsystem SUBS1 can now be considered as an element of the system SUBS3 and identified with an element code of SUBS1. The element mode probabilities associated with SUBS1 are available on the disk files. This information is conveyed to the simulator by means of the S code of 0 for the element designated by the element code SUBS1 in the SUBS3 data set. The SUBS3 and SUBS2 subsystems are processed according to the input S codes for their elements, leading to a final SYSTEM reliability simulation.

The alternate execution of the simulation and analysis phases of the RAPID technique is necessitated by the limited size of the core memory available on the 1620 system, since a "nested" subroutine technique would require all the subroutines containing models to reside in core memory at the same time.

3.3.5 System Outputs

The IBM 1620 Reliability Simulator has a group of subroutines which output the results of a system simulation and analysis in a set format. The pertinent input data and system identification are provided along with the results, for easy reference. The system model is printed also, if this is requested at the time the input data are submitted for running on the computer. This print-out of the model is accomplished by setting a switch on the IBM 1620 console.

To illustrate the standard output format, the three-resistor Basic Example Network, Figure B-1 in Section 2, was analyzed using the Reliability Simulator. The output from this computer run is given in Figures B-42 and B-43. Note that the general form of the model in Figure B-42 follows the one developed in Section 2.3, with the exception that the variable "X" is employed in the figure as an intermediate step in developing the Total Row Probability statements.

The first part of Figure B-43* summarizes the SDA and SMA inputs under the standard title block providing the system name and system mode name. Each submode of the SMA is numbered in increasing order. If more than one submode exists within a given order, these are numbered sequentially also, as noted under the "(O/N)" column of the output SMA.

The model must be evaluated for each point in time for which a reliability value is desired. The TIME CARDS in the input deck define those points in time to the simulator. After the SDA and SMA inputs are given, outputs are given for each point in time summarizing the probability calculation involved in the model. The time value is given followed by columns identifying:

a. The Submode Probabilities (called "Submode Probability" on the page),

b. A factor called Relative Weight, indicating the proportion of the total sum of Submode Probabilities contained in each submode,

c. The sum of the total Row Probabilities within each order,

d. The sum of all Total Row Probabilities, i.e. P(SYSTEM OPEN MODE).

The total mode summary table follows the summaries of the calculations for each point in time. This is simply a review of previous

---

*In order to conserve space, data in this figure, which ordinarily would be printed on separate pages, has been combined.

calculations with an additional column for one minus the system mode probability.

The output format concludes with Element Descriptions (ordinarily one page per element) summarizing the element data originally provided as input to the Simulator.

The element summaries for the three-resistor example system are given on the last three sections of Figure B-43.

```
SUBROUTINE RAP016 (BP,TR,P)
DIMENSION BP(100),TR(100),P(50,10)
BP(1)=P(1,2)
BP(2)=P(2,2)*P(3,2)
X=1.
TR(1)=X
TR(1)=TR(1)*BP(1)
X=(1.-P(1,2))
TR(2)=X
TR(2)=TR(2)*BP(2)
RETURN
END
```

Figure B-42. Reliability Model for 3-Resistor Basic Example Network

```
RAPID-M1           * PROBABILISTIC SYSTEM MODE ANALYSIS *      PAGE  1 OF 12
RES-SYS                 EXAMPLE RESISTOR SYSTEM                DATE  9/14/65
                           SYSTEM OPEN MODE
--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--
                               SYSTEM
                           DEFINITION ARRAY

                  --------------------ELEMENTS--------------------
                           1         2         3         4         5
                  12345678901234567890123456789012345678901234567890
                **                                                  **
  SPECIFIED MODES * 222                                              *
                **                                                  **

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--
                               SYSTEM
                             MODE ARRAY

                  --------------------ELEMENTS--------------------
                           1         2         3         4         5
                  12345678901234567890123456789012345678901234567890
  SUBMODE ( 0/ N) **                                                **
     1    ( 1/ 1) * 1                                                *
     2    ( 2/ 1) *  11                                              *
                **                                                  **

--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--
```

TIME = 0.00000 HOURS

SUBMODE	SUBMODE (0/ N)	SUBMODE PROBABILITY	RELATIVE WEIGHT	TOTAL ORDER PROBABILITY	TOTAL MODE PROBABILITY
1	(1/ 1)	0.00000000		0.00000000	
2	(2/ 1)	0.00000000		0.00000000	0.00000000

--*--

TIME = 1.00000 HOURS

SUBMODE	SUBMODE (0/ N)	SUBMODE PROBABILITY	RELATIVE WEIGHT	TOTAL ORDER PROBABILITY	TOTAL MODE PROBABILITY
1	(1/ 1)	.00008999	.9999100	.00008999	
2	(2/ 1)	0.00000000	.0000899	0.00000000	.00009000

--*--

TIME = 2.00000 HOURS

SUBMODE	SUBMODE (0/ N)	SUBMODE PROBABILITY	RELATIVE WEIGHT	TOTAL ORDER PROBABILITY	TOTAL MODE PROBABILITY
1	(1/ 1)	.00017998	.9998200	.00017998	
2	(2/ 1)	.00000003	.0001799	.00000003	.00018001

Figure B-43. Standard Simulator Output Format (1 of 3)

--*--

TIME = 3.00000 HOURS

SUBMODE	SUBMODE (O/ N)	SUBMODE PROBABILITY	RELATIVE WEIGHT	TOTAL ORDER PROBABILITY	TOTAL MODE PROBABILITY
1	(1/ 1)	.00026995	.9997301	.00026995	
2	(2/ 1)	.00000007	.0002698	.00000007	.00027002

--*--

TIME = 3.26666 HOURS

SUBMODE	SUBMODE (O/ N)	SUBMODE PROBABILITY	RELATIVE WEIGHT	TOTAL ORDER PROBABILITY	TOTAL MODE PROBABILITY
1	(1/ 1)	.00029394	.9997061	.00029394	
2	(2/ 1)	.00000008	.0002938	.00000008	.00029403

--*--

TIME = 3.53333 HOURS

SUBMODE	SUBMODE (O/ N)	SUBMODE PROBABILITY	RELATIVE WEIGHT	TOTAL ORDER PROBABILITY	TOTAL MODE PROBABILITY
1	(1/ 1)	.00031793	.9996821	.00031793	
2	(2/ 1)	.00000010	.0003178	.00000010	.00031804

--*--

TIME = 3.80000 HOURS

SUBMODE	SUBMODE (O/ N)	SUBMODE PROBABILITY	RELATIVE WEIGHT	TOTAL ORDER PROBABILITY	TOTAL MODE PROBABILITY
1	(1/ 1)	.00034192	.9996582	.00034192	
2	(2/ 1)	.00000011	.0003418	.00000011	.00034204

--*--

TOTAL MODE SUMMARY TABLE

TIME (HOURS)	OCCURENCE PROBABILITY	N-OCCURENCE PROBABILITY
0.000000E-99*	0.00000000	1.00000000
1.000000E+00	.00009000	.99991000
2.000000E+00	.00018001	.99981999
3.000000E+00*	.00027002	.99972998
3.266666E+00	.00029403	.99970597
3.533333E+00	.00031804	.99968196
3.800000E+00*	.00034204	.99965796

NOTE * INDICATES A TRANSITION TIME

Figure B-43. Standard Simulator Output Format (2 of 3)

--*--

ELEMENT DESCRIPTION
ELEMENT-01
RESISTOR 1 E1

FAILURE RATE PER MILLION HOURS = 100.00000

ELEMENT MODE SYMBOL	ELEMENT MODE NAME	CONDITIONAL PROBABILITY
0	---	---
1	OPEN	.9000
2	SHORT	.1000

TIME INTERVAL (HOURS)	K FACTOR
0.000E-99 TO 3.000E+00	1.00E+00
3.000E+00 TO 3.800E+00	1.00E+00

--*--

ELEMENT DESCRIPTION
ELEMENT-02
RESISTOR 2 E2

FAILURE RATE PER MILLION HOURS = 100.00000

ELEMENT MODE SYMBOL	ELEMENT MODE NAME	CONDITIONAL PROBABILITY
0	---	---
1	OPEN	.9000
2	SHORT	.1000

TIME INTERVAL (HOURS)	K FACTOR
0.000E-99 TO 3.000E+00	1.00E+00
3.000E+00 TO 3.800E+00	1.00E+00

--*--

ELEMENT DESCRIPTION
ELEMENT-03
RESISTOR 3 E3

FAILURE RATE PER MILLION HOURS = 100.00000

ELEMENT MODE SYMBOL	ELEMENT MODE NAME	CONDITIONAL PROBABILITY
0	---	---
1	OPEN	.9000
2	SHORT	.1000

TIME INTERVAL (HOURS)	K FACTOR
0.000E-99 TO 3.000E+00	1.00E+00
3.000E+00 TO 3.800E+00	1.00E+00

Figure B-43. Standard Simulator Output Format (3 of 3)

3.3.6 Program*

The program which implements the RAPID technique on the IBM 1620 is designed to operate under the LSI version of IBM's 1620-1443 Monitor II. Because of the size of the program, it has been coded as two mainline programs referred to as routines which link to each other for the required transition from the simulation phase to the analysis phase and vice-versa. These two routines and their required subroutines are presented in a brief narrative, flow chart, and program listing form in this section. The only exception to this is the subroutine RAP016, which is generated by the Reliability Simulator and contains the system reliability model. Because this subroutine is different for each model generated, only a typical program listing and a brief narrative are included in this section. The assembler language used is SPS-IID and the higher level source language is FORTRAN II-D.

There are some limitations on the inputs to the IBM 1620 version of the RAPID technique. These represent limitations on the inputs only. It is possible to keep the inputs within these limits and have the size of the generated subroutine RAP016 large enough to prohibit compilation or execution of the analysis phase. If this occurs, it is suggested to divide the system into subsystems and repeat the simulation and analysis.

- Number of Elements - One to fifty elements are allowed for each system.
- Element Modes - One non-failure mode and from one to nine failure modes are are allowed for each element.
- System Modes - One non-failure mode and from one to nine failure modes are allowed for each system or subsystem.
- Submodes - From one to one-hundred rows or submodes are allowed for each System Mode Array.
- Transition Times - From one to twenty-five transition times are allowed for a given analysis.

*All figures mentioned in this section will be found in the end of this section.

RAPID1 (Figures B-44 and B-45)*

This routine is one of two "mainline" programs which direct the RAPID simulation and analysis. The data input and model generation are carried out by this routine and its subroutines. RAPID1 determines which inputs are required for the simulation and analysis and calls on the appropriate subroutines. The CALL MONITR statement directs the FORTRAN II-D Compiler to compile the subroutine RAP016 which was coded by RAP004 and store it in object form on the disk files. The CALL LINK (RAPID2) statement causes the second "mainline" program, RAPID2, to be loaded and executed.

RAPID2 (Figures B-46 and B-47)

This routine is the "mainline" program which directs the analysis portion of RAPID. The CALL LINK (RAPID1) statement causes the first "mainline" program, RAPID1, to be loaded and executed.

RAP001 (Figures B-48 and B-49)

This subroutine is called by the routine RAPID1. RAP001 will read in the type 1,2,3 amd 4 cards and check these cards for errors. The sequence of cards is also checked. The System Definition Array is checked for numeric entries and also determines if the proper number of element entries are present. The table of system code and mode numbers is scanned to determine if there has already been a similarly designated system analyzed. The location of this table and other work areas of the disk referred to below are given in Table B-1.

---

*Figures B-44 through B-87 represent Logic Flow Charts and Program Listings for all of the IBM 1620 programs and subprograms.

Table B-1. Disk Storage Layout, RAP001 - RAP009

RECORD NOS.	CONTENTS	INDEX*
1-500	C(I, J), E(I, J)	10xI+J-9
501-1500	ELEMENT MODE NAMES	490+10xI+J
1501-1550	ELEMENT NAMES	1500+I
1551-1600	ELEMENT CODES	1550+I
1601-1850	K FACTORS	1596+5xI
1851-2050	CURRENT SUMMARY TABLE	
2051-2200	STORED SYSTEM CODES	
2201-9999	STORED PROBABILITIES	

*I = ELEMENT NUMBER

J = ELEMENT MODE NUMBER

The indicator IERROR conveys the reason for entry to RAP001. IERROR equal to zero indicates a new system is being analyzed and the first card read should be a type 1 card. If IERROR is equal to one, then an error had been detected in a data set and RAP001 must locate the beginning of the next data set. This is accomplished by means of the FRSTRD subroutine. This subroutine enables column 80 of each input card to be examined, before the information on the card is transmitted to computer memory, until a type 1 card or the end of the input data is encountered.

RAP002 (Figures B-50 and B-51)

This subroutine is called by the routine RAPID1. This subroutine reads in the type 5 cards which contain the System Mode Array. Each card is checked for valid entries. If an input error is detected, IERROR is set equal to one and a branch back to RAPID1 is executed.

If there are no errors in the input, the System Mode Array is rearranged so that first order submodes appear first, second order submodes are next, and so on. RAP002 also determines the population of submodes within each order. The rearranged System Mode Array is then stored on the disk file for use by other subroutines.

RAP003 (Figures B-52 and B-53)

This subroutine is called by the routine RAPID1. RAP003 performs the STRIP logic to take into account the fact that some submodes of the System Mode Array may overlap one another. The System Mode Array constructed by RAP015 is processed. This STRIPed System Mode Array is then stored on the disk file for processing by subroutine RAP004.

RAP004 (Figures B-54 and B-55)

This subroutine is called by the routine RAPID1. RAP004 utilizes the coded information stored by RAP003 on the disk file and generates the subroutine RAP016, written in FORTRAN II-D. RAP016 is stored in card image format (2 sectors per card) on the disk files.

RAP005 (Figures B-56 and B-57)

This subroutine is called by the routine RAPID1. RAP005 determines the total number of pages of output. This is calculated as a function of total number of points in time for which a detailed analysis is required, the number of rows in the System Mode Array, and the number of elements in the system. The first page of output is also printed by this subroutine. If the System Mode Array contains more than 35 rows, a second page is printed. A third page is required if the System Mode Array contains more than 81 rows.

RAP007 (Figures B-58 and B-59)

This subroutine is called by the routine RAPID1. RAP007 reads and checks the type 6 cards. If an S code for an element is zero, a search for the stored probabilities is made to locate them. If they are not found, IERROR is set to one and a branch back to RAPID1 is executed. For an S code of one, a check on the conditional probabilities is made to determine if they sum to one. A non-one sum indicates an error in the input data.

Element mode numbers and element numbers are also checked for validity. Any error condition will result in a diagnostic message on the printer, IERROR set equal to one, and a branch back to RAPID1.

RAP008 (Figures B-60 and B-61)

This subroutine is called by the routine RAPID1, if any element of the system under study has an S code of one. This subroutine reads in type 7 cards which contain the environmental (K) factors to be used with an exponential failure distribution. RAP008 then records these values on the disk file for later use in the analysis.

The element numbers are checked for validity and the sequencing of the data cards is also checked. A check is also made to determine whether all required K factors have been supplied or if too many type 7 cards have been included. In the event an error is detected, IERROR is set equal to one and a diagnostic message is printed. A branch back to RAPID1 is then executed, returning control to that routine.

RAP009 (Figures B-62 and B-63)

This subroutine is called by the routine RAPID1 if any element of the system under study has an S code of two, indicating that probabilities for that element are to be read in from type 8 cards. RAP009 records these element probabilities on the disk files and makes corresponding entries in the table of recorded probabilities. After this is accomplished, these elements are treated in exactly the same manner as those elements with an S code of zero.

The element numbers are checked for validity along with the sequencing of data cards. A check is also made to determine if sufficient type 8 cards have been supplied. In the event an error is detected, IERROR is set equal to one and a diagnostic message printed. A branch back to RAPID1 is executed, returning control to that routine.

RAP010 (Figures B-64 andB-65)

This subroutine is called by the routine RAPID2. RAP010 determines the value of the required element mode probabilities. For elements with an S code of zero or two, the probabilities are read from the disk file. For elements with an S code equal to one, the exponential failure rate information is used to calculate the element mode probabilities for time t.

RAP011 (Figures B-66 and B-67)

This subroutine is used in conjunction with RAP010 and is called by the routine RAPID2 if any of the elements require their element mode probabilities to be calculated from the exponential function.

In effect, this subroutine carries out the task of summing the factors that go into the x of e^{-x} . An updating of this sum is performed at each transition time. This sum is recorded on the disk files and used by RAP010 as the variable E.

RAP012 (Figures B-68 and B-69)

This subroutine is called by the routine RAPID2. RAP012 prints out a detailed analysis page for each time t that the system reliability is computed. The submode relative weights and total order probabilities are computed within this subroutine. The probability of occurrence of the system mode is recorded on the disk files along with the probability of non-occurrence.

RAP013 (Figures B-70 and B-71)

This subroutine is called by the routine RAPID2. RAP013 prints out a summary table of the probabilities of occurrence and non-occurrence of the system mode under consideration. The number of pages printed for the table is determined by this subroutine, and appropriate continuation page headings are included when necessary.

Upon completion of the printing, the probabilities are recorded on the disk files and an entry in the table of stored probabilities is made. These probabilities can be used as element mode probabilities in later analyses by specifying an element code to be the same as the current system code and by giving the element an S code of zero.

RAP014 (Figures B-72 and B-73)

This subroutine is called by the routine RAPID2. RAP014 prints a one page description for each element in the system. This description contains the element number and its code. The exponential failure rate per million hours is also included for those elements with an S code of one along with environmental (K) factors for the various time intervals. The element modes available to each element and, if supplied, element mode names are also printed on the element description page.

RAP015 (Figures B-74 and B-75)

This subroutine may be called by the routine RAPID1. A system code which is non-zero indicates the System Mode Array must be stored on the disk files for possible combination with other System Mode Arrays of the same system. RAP015 is the subroutine which combines these System Mode Arrays. The variable IBUILD indicates the number of submodes in the combined System Mode Array.

RAP016 (Figure B-42)

Subroutine RAP016 is written in FORTRAN II by subroutine RAP004 and compiled by a call to the Monitor II system from RAPID1. It contains the model or equation for the system mode being analyzed by RAPID2. Since this subroutine is different for each System Mode Array, the listing which appears in this appendix is only typical.

RAP017 (Figures B-76 and B-77)

Subroutine RAP017 is called by the routine RAPID1 if the value of MCODE is zero. This subroutine then takes the combined System Mode Array stored on disks and rearranges it as one large SMA. The rearrangement consists of placing first order submodes, if present, as the first rows of the SMA, then second order submodes and so on. The subroutine then defines the number of rows. It also counts the number of submodes of each order. This information is recorded on disks as NUMOR (1) through NUMOR (50). The location of the NUMOR array on disk is given in Table B-2.

Table B-2. Disk Storage Layout RAP017 - RAP021

SECTOR	VARIABLES	TOTAL DIGITS
110000	NUMOR(1)...NUMOR(50)	100
110001	NST(1)...NST(25)	100
110002	NST(26)...NST(50)	100
110003	ICODE(1)...ICODE(25)	100
110004	ICODE(26)...ICODE(50)	100
110005	NIT(1)...NIT(25)	100
110006	FLAMB(1)... FLAMB(10)	100
110007	FLAMB(11)... FLAMB(20)	100
110008	FLAMB(21)... FLAMB(30)	100
110009	FLAMB(31)... FLAMB(40)	100
110010	FLAMB(41)... FLAMB(50)	100
110011	LN(1)... LN(23), NT, IPAGE	100
110012	LM(1)... LM(23), NEL, ITOTAL	100
110013	TT(1)... TT(10)	100
110014	TT(11)... TT(20)	100
110015	TT(21)... TT(25), LS(1)... LS (8), LD(1)... LD(3), NR	98
110016	MENT, ITAB, IAVAIL, MCODE, IBUILD, MSAVE(1)... MSAVE(4)	36
110100 . . 110199	CURRENT SYSTEM MODE ARRAY	
110200 . . . 111099	COMBINED SYSTEM MODE ARRAY	

RAP018 (Figures B-78 and B-79)

The subroutine RAP018 is called by the routine RAPID1. The purpose of the subroutine is to record the values of the following variables and arrays on disks.

ARRAYS (SIZE)	VARIABLES
NST (50)	NEL
ICODE (50)	NT
NIT (25	MENT
FLAMB (50)	ITAB
LN(23)	IAVAIL
LM (23)	IPAGE
LD (3)	ITOTAL
LS (8)	NR
TT (25)	MCODE
MSAVE (4)	IBUILD

The arrangement of these variables and arrays along with disk sector addresses is given in Table B-2.

RAP019 (Figures B-80 and B-81)

The subroutine RAP019 is called by the routine RAPID2. The purpose of the subroutine is to set the values of the following arrays and variables according to values stored on disks.

ARRAYS (SIZE)	VARIABLES
NST (50)	NEL
ICODE (50)	NT
NIT (25)	MENT
FLAMB (50)	ITAB
LN (23)	IAVAIL
LM (23)	IPAGE
LD (3)	ITOTAL
LS (8)	NR
TT (25)	MCODE
MSAVE (4)	IBUILD

The arrangement of these variables and arrays along with disk sector addresses is given in Table B-2.

RAP020 (Figures B-82 and B-83)

Subroutine RAP020 is called by the routine RAPID2. RAP020 records the values of the following variables on disk.

MENT	MSAVE (1)
ITAB	MSAVE (2)
IAVAIL	MSAVE (3)
IBUILD	MSAVE (4)

The locations on disks, including sector addresses, where these variables are recorded are given in Table B-2.

RAP021 (Figures B-84 and B-85)

Subroutine RAP021 is called by the routine RAPID1. This subroutine reads the values for the following variables from disks and sets variables used in RAPID1 accordingly.

MENT	MSAVE (1)
ITAB	MSAVE (2)
IAVAIL	MSAVE (3)
IBUILD	MSAVE (4)

The locations on disk, including sector addresses, where these variables are recorded are given in Table B-2.

RAP022 (Figures B-86 and B-87)

This routine initializes all tables and indicators for the RAPID analysis. The variable MENT is set to zero, indicating there are no probabilities stored on the disks. ITAB is set equal to 1851, indicating there are no entries in the summary table. IAVAIL is set equal to 2201, which is the first record number available for storing calculated probabilities. MCODE and IBUILD are both set equal to zero to indicate there has not been a previous analysis for the current system and as a result, there is no combined SMA. MSAVE (1) through MSAVE (4) are also zeroed out to prevent a possible error condition the first time RAPID1 uses them.

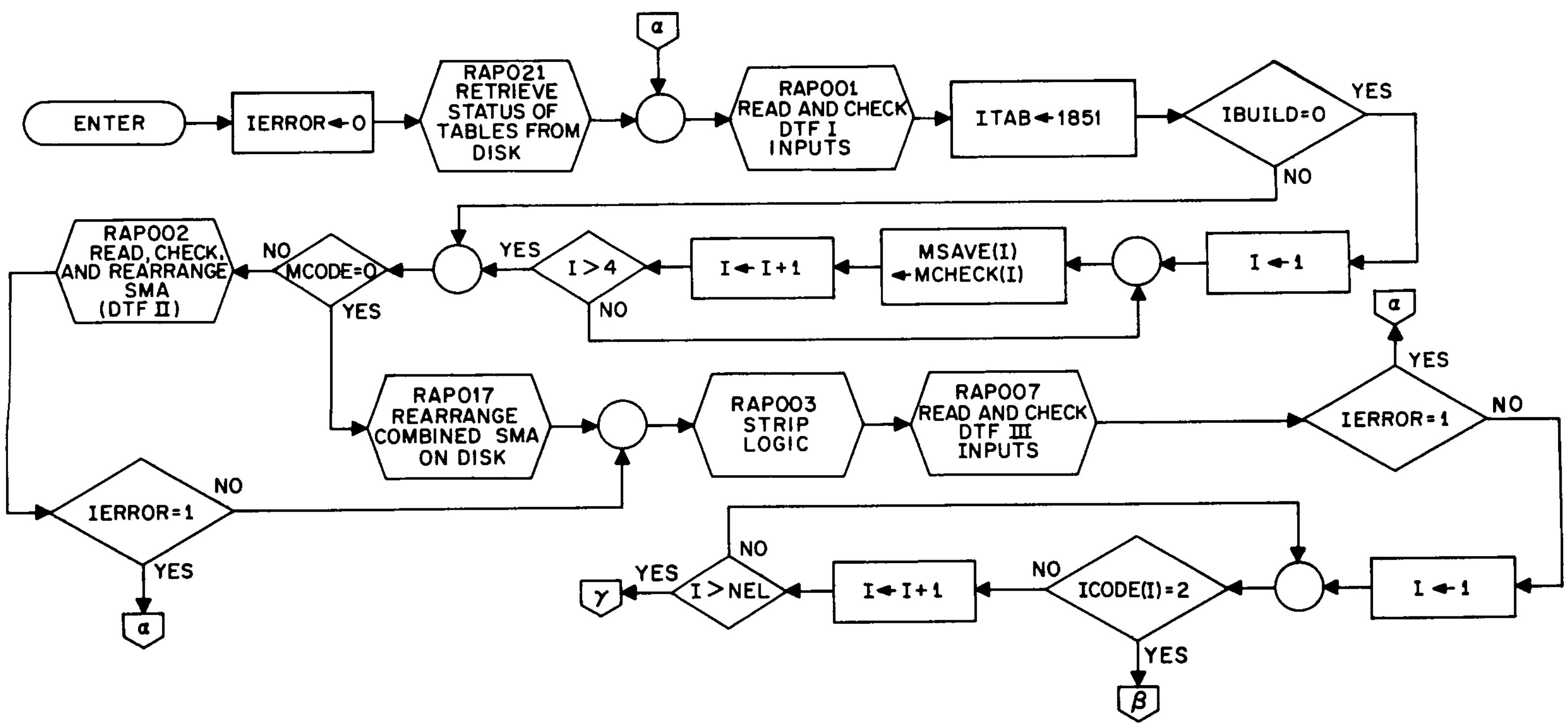


Figure B-44. RAPID1 Mainline Program, Logic Flow Chart (1 of 2)

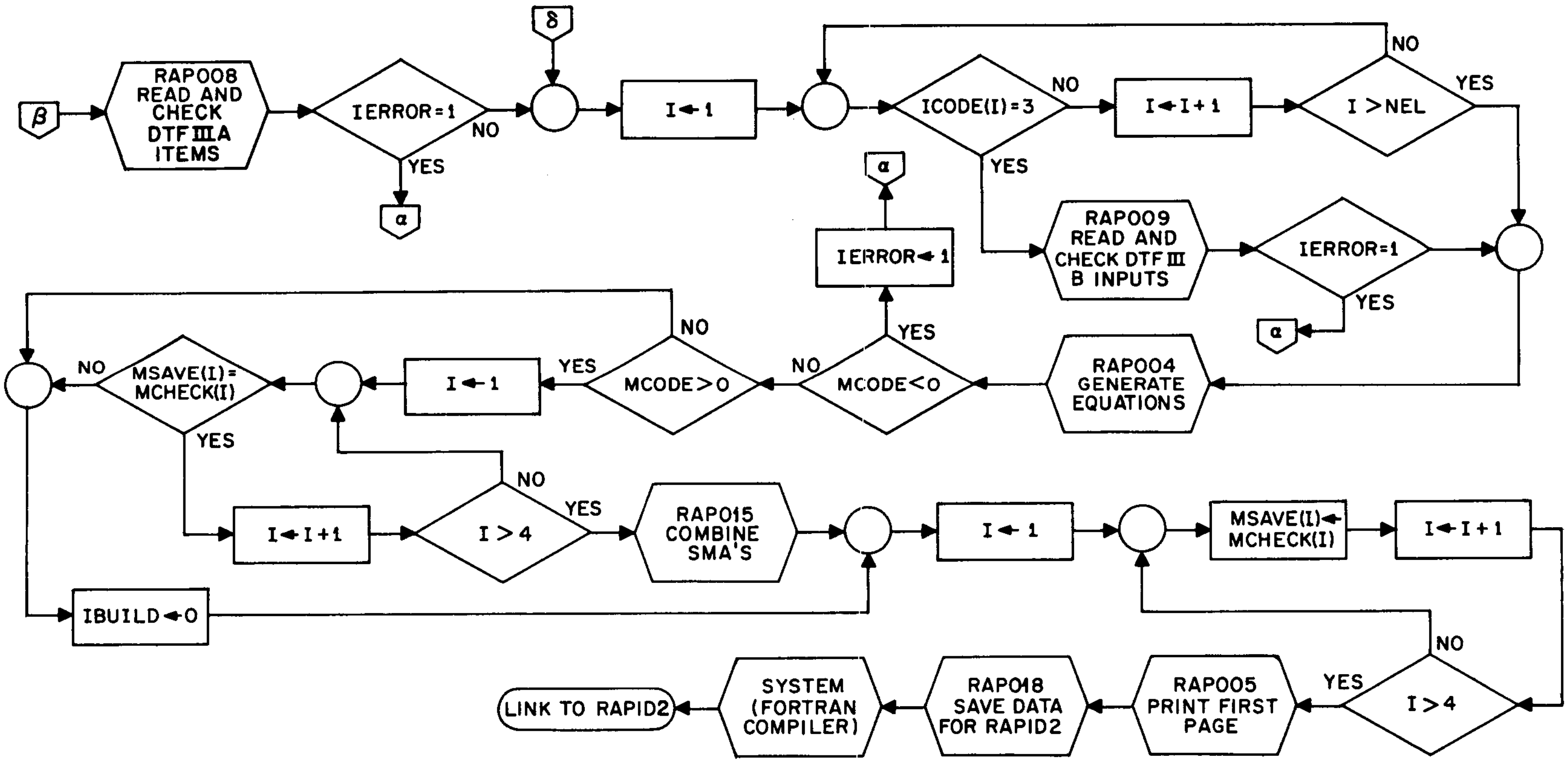


Figure B-44. RAPID1 Mainline Program, Logic Flow Chart (2 of 2)

```
**       RAPID1
C
C          RAPID1 - - - MAINLINE PROGRAM
C
C     FLAMB(50)    ARRAY CONTAINING FAILURE RATES PER HOUR
C     I            INDEX
C     IAVAIL       CURRENT AVAILABLE RECORD NUMBER
C     IBUILD       NUMBER OF ROWS IN COMBINED SMA
C     ICODE(50)    ARRAY CONTAINING S CODE
C     IERROR       INDICATOR, 0 IF NO INPUT ERROR, 1 IF AN INPUT ERROR OCCURRED
C     IPAGE        PAGE NUMBER
C     ITAB         INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
C     ITOTAL       TOTAL NUMBER OF PAGES
C     L            INDEX
C     LD(3)        DATE OF RUN
C     LM(23)       SYSTEM MODE NAME
C     LN(23)       SYSTEM NAME
C     LS(8)        SYSTEM CODE
C     MA           DUMMY VARIABLE
C     MCHECK(4)    SYSTEM CODE SAVED IN THIS ARRAY
C     MCODE        SYSTEM MODE NUMBER
C     MENT         NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
C     MSAVE(4)     SYSTEM CODE SAVED IN THIS ARRAY
C     NARRAY       NUMBER OF PROBABILITY ARRAYS
C     NEL          NUMBER OF ELEMENTS IN SYSTEM
C     NEX          NUMBER OF SPACING LINES TO APPEAR ON INTERMEDIATE
C                  PRINT OUTS
C     NIT(25)      NUMBER OF INTERMEDIATE TIMES FOR EACH TRANSITION INTERVAL
C     NR           NUMBER OF ROWS IN SYSTEM MODE ARRAY
C     NST(50)      ENTRIES OF THE SYSTEM DEFINITION ARRAY
C     NT           NUMBER OF TRANSITION TIMES
C     SECT         NUMBER OF SECTORS PER ARRAY
C     TT(25)       TRANSITION TIMES
C
      DIMENSION LS(8),LN(23),LD(3),LM(23),NST(50),TT(25),NIT(25),ICODE(5
     10),FLAMB(50)
      DIMENSION MSAVE(4),MCHECK(4)
      DIMENSION MA(1,1)
      DEFINE DISK (8,9999)
      IERROR=0
      L=1
      MA(L,L)=L
      FIND (L)
      CALL RAP021 (MENT,ITAB,IAVAIL,IBUILD,MSAVE(1))
    1 CALL RAP001 (LS,LN,LD,MCODE,LM,NEL,NST,TT,NIT,MENT,NT,IERROR,MCHEC
     1K)
      ITAB=1851
```

Figure B-45. RAPID1 Mainline Program, Program Listing (1 of 2)

```
      IF(IBUILD)20,18,20
   18 DO 19 I=1,4
   19 MSAVE(I)=MCHECK(I)
   20 CONTINUE
      IF(MCODE)22,21,22
   21 CALL RAP017 (IBUILD,NEL,NR)
      GO TO 3
   22 CALL RAP002(NEL,NST,NEX,NR,IERROR)
      IF(IERROR-1)3,1,3
    3 CALL RAP003(NEL,NST,NR,NARRAY,SECT,MCODE)
      CALL RAP007 (NEL,NST,ICODE,FLAMB,MENT,IERROR)
      IF(IERROR-1)4,1,4
    4 DO 5 I=1,NEL
      IF(ICODE(I)-2)5,6,5
    5 CONTINUE
      GO TO 7
    6 CALL RAP008 (NEL,NT,ICODE,IERROR)
      IF(IERROR-1)7,1,7
    7 DO 8 I=1,NEL
      IF(ICODE(I)-3)8,9,8
    8 CONTINUE
      GO TO 10
    9 CALL RAP009 (ICODE,NEL,NST,NT,NIT,IAVAIL,MENT,IERROR)
      IF(IERROR-1)10,1,10
   10 CALL RAP004 (NEL,NST,NARRAY,SECT,NR,MCODE)
      IF(MCODE)12,15,13
   12 IERROR=1
      GO TO 1
   13 DO 14 I=1,4
      IF(MSAVE(I)-MCHECK(I))15,14,15
   14 CONTINUE
      CALL RAP015(NR,IBUILD)
      GO TO 16
   15 IBUILD=0
   16 DO 17 I=1,4
   17 MSAVE(I)=MCHECK(I)
      CALL RAP005 (LS,LN,LD,MCODE,LM,NST,NEL,NEX,NR,IPAGE,ITOTAL,NT,NIT)
      CALL RAP018 (NST,ICODE,NIT,FLAMB,LN,LM,LS,LD,NT,NEL,TT,MENT,ITAB,I
     1AVAIL,IPAGE,ITOTAL,NR,MCODE,IBUILD,MSAVE(1))
      CALL MONITR
      CALL LINK(RAPID2)
   11 L=FRSTRD(L)
      END
```

Figure B-45. RAPID1 Mainline Program, Program Listing (2 of 2)

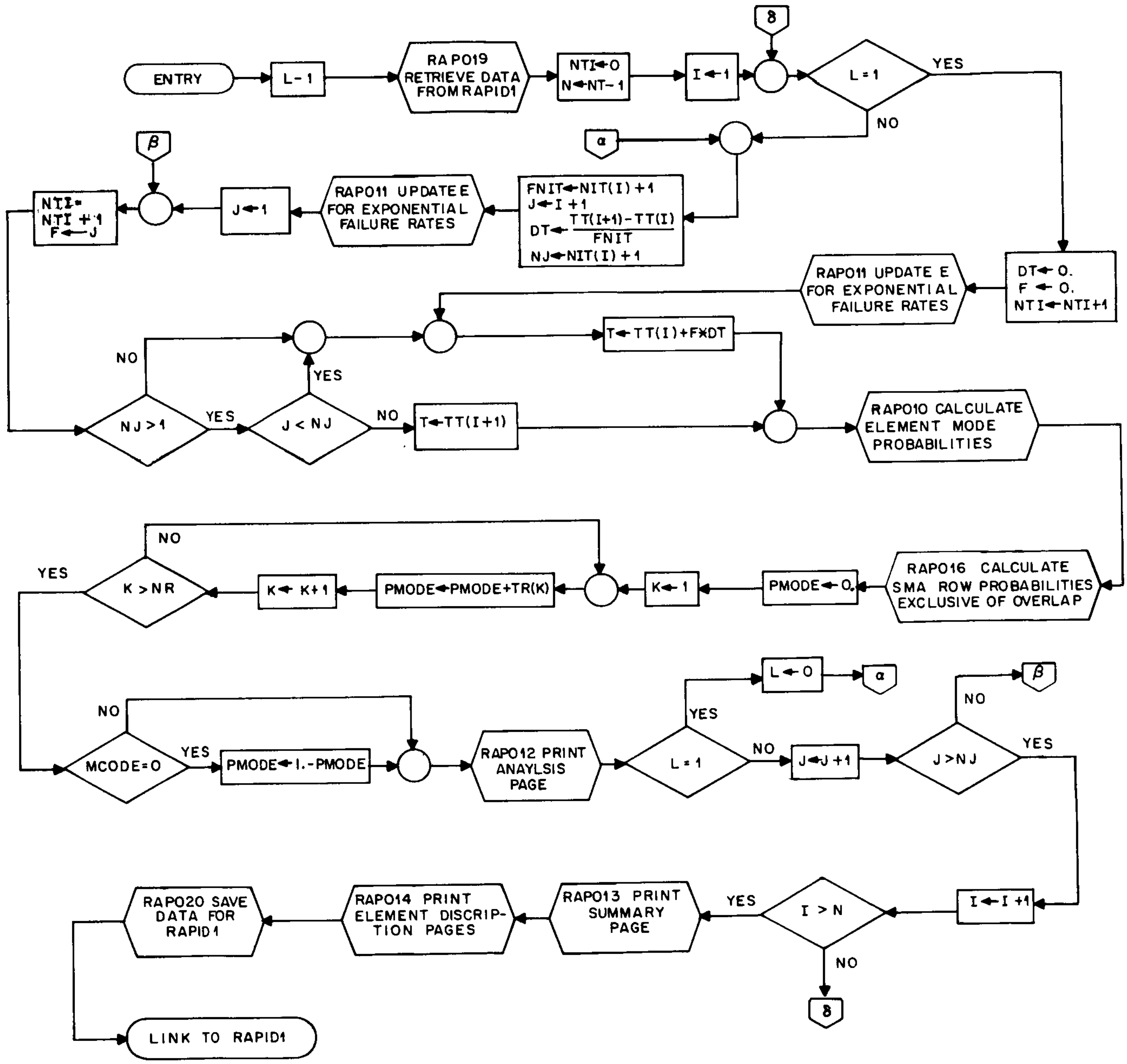


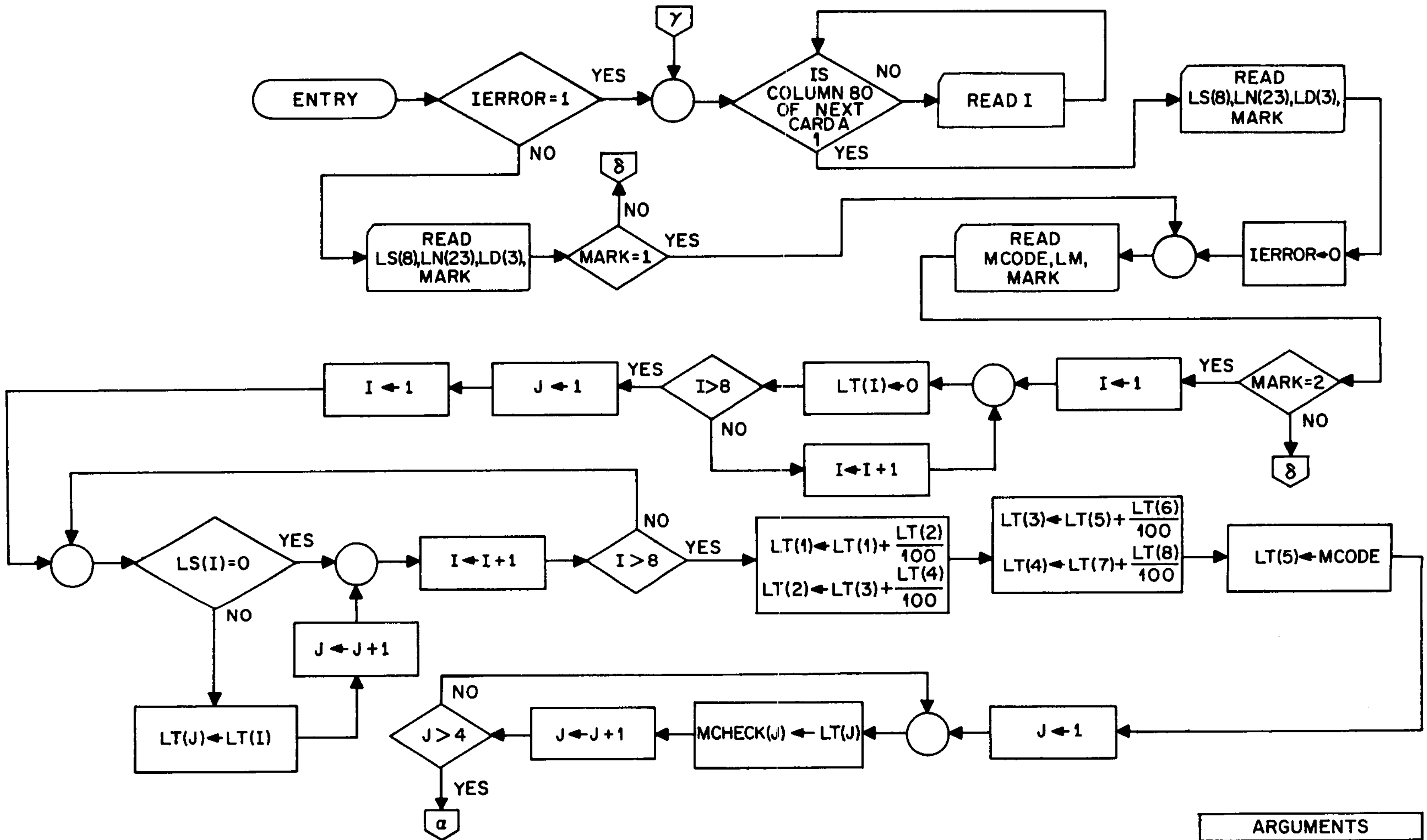
Figure B-46. RAPID2 Mainline Program, Logic Flow Chart

```
C
C          RAPID2 - - - MAINLINE PROGRAM
C
C     BP(100)       BASIC SUBMODE PROBABILITIES
C     DT            TIME INTERVAL
C     F             WORKING VARIABLE
C     FLAMB(50)     ARRAY CONTAINING FAILURE RATES PER HOUR
C     FNIT          WORKING VARIABLE
C     I             INDEX
C     IAVAIL        CURRENT AVAILABLE RECORD NUMBER
C     IBUILD        NUMBER OF ROWS IN COMBINED SMA
C     ICODE(50)     ARRAY CONTAINING S CODE
C     IPAGE         PAGE NUMBER
C     ITAB          INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
C     ITOTAL        TOTAL NUMBER OF PAGES
C     J             INDEX
C     K             INDEX
C     L             INDEX
C     LD(3)         DATE OF RUN
C     LM(23)        SYSTEM MODE NAME
C     LN(23)        SYSTEM NAME
C     LS(8)         SYSTEM CODE
C     MCODE         SYSTEM MODE NUMBER
C     MENT          NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
C     MSAVE(4)      SYSTEM CODE SAVED IN THIS ARRAY
C     N             NUMBER OF TRANSITION INTERVALS
C     NEL           NUMBER OF ELEMENTS IN SYSTEM
C     NIT(25)       NUMBER OF INTERMEDIATE TIMES FOR EACH TRANSITION INTERVAL
C     NJ            NUMBER OF ELEMENT MODES FOR CURRENT ELEMENT
C     NR            NUMBER OF ROWS IN SYSTEM MODE ARRAY
C     NST(50)       ENTRIES OF THE SYSTEM DEFINITION ARRAY
C     NT            NUMBER OF TRANSITION TIMES
C     NTI           TIME INDEX FOR STORED PROBABILITIES
C     NUMOR(50)     ARRAY CONTAINING NUMBER OF SUBMODES OF EACH ORDER
C     P(50,10)      ELEMENT MODE PROBABILITY
C     PMODE         PROBABILITY OF SYSTEM MODE OCCURRING
C     T             CURRENT TIME
C     TR(100)       SUBMODE PROBABILITIES EXCLUSIVE OF OVERLAP
C     TT(25)        TRANSITION TIMES
C
      DIMENSION  LS(8),LN(23),LD(3),LM(23),NST(50),TT(50),NIT(25),ICODE(
     150),FLAMB(50),TR(100),BP(100),NUMOR(50),P(50,10)
      DIMENSION MSAVE(4)
      DEFINE DISK (8,9999)
      F=0.
      L=EXPF(F)
      FIND (L)
      P(L,L)=1.
      CALL RAPO19 (NST,ICODE,NIT,FLAMB,LN,LM,LS,LD,NT,NEL,TT,MENT,ITAB,I
     1AVAIL,IPAGE,ITOTAL,NR,MCODE,IBUILD,NUMOR,MSAVE(1))
      NTI=0
      N=NT-1
      DO 8 I=1,N
      IF (L-1)2,1,2
    1 DT=0.
      F=0.
      NTI=NTI+1
      CALL RAPO11(NEL,NST,TT,I,FLAMB,NT,ICODE)
      GO TO 3
    2 FNIT=NIT(I)+1
```

Figure B-47.   RAPID2 Mainline Program, Program Listing (1 of 2)

```
      J=I+1
      DT=(TT(J)-TT(I))/FNIT
      NJ=NIT(I)+1
      CALL RAP011(NEL,NST,TT,I,FLAMB,NT,ICODE)
      DO 7 J=1,NJ
      NTI=NTI+1
      F=J
      IF(NJ-1)3,3,20
   20 IF(J-NJ)3,4,4
    3 T=TT(I)+F*DT
      GO TO 21
    4 T=TT(I+1)
   21 CALL RAP010 (NEL,NST,T,TT,FLAMB,NT,ICODE,NTI,MENT,I,P)
      CALL RAP016(BP,TR,P)
      PMODE=0.
      DO 10 K=1,NR
   10 PMODE=PMODE+TR(K)
      IF(MCODE)12,11,12
   11 PMODE=(.99999999 -PMODE)+.00000001
   12 CALL RAP012 (BP,IPAGE,ITOTAL,LS,LN,LD,LM,T,NR,ITAB,TR,NUMOR,NEL,PM
     1ODE,MCODE)
      IF (L-1)7,9,7
    9 L=0
      GO TO 2
    7 CONTINUE
    8 CONTINUE
      CALL RAP013 (ITAB,MENT,IPAGE,ITOTAL,LS,LN,LD,LM,IAVAIL,TT,NT,MCODE
     1)
      CALL RAP014 (IPAGE,ITOTAL,LS,LN,LD,LM,NEL,NST,ICODE,FLAMB,TT,NT)
      CALL RAP020 (MENT,ITAB,IAVAIL,IBUILD,MSAVE(1))
      CALL LINK(RAPID1)
      END
```

Figure B-47. RAPID2 Mainline Program, Program Listing (2 of 2)



ARGUMENTS	
INPUT	OUTPUT
IERROR	LD
MENT	LM
	LN
	LS
	MCHECK
	MCODE
	NEL
	NIT
	NST
	NT
	TT

Figure B-48. RAP001 Subprogram, Logic Flow Chart (1 of 2)

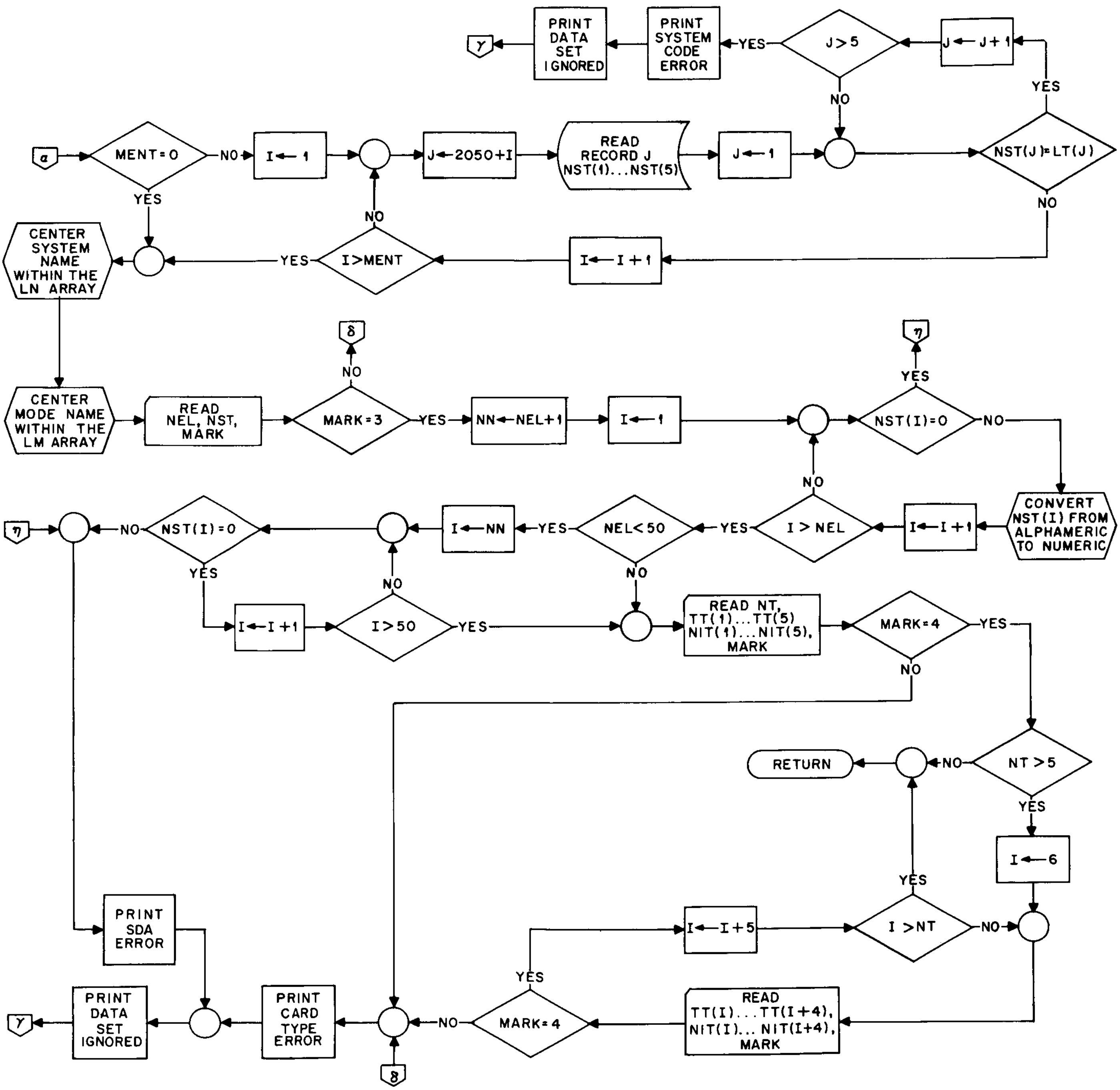


Figure B-48. RAP001 Subprogram, Logic Flow Chart (2 of 2)

```
C
C        RAP001 - - - INPUT FROM DATA TRANSMITTAL FORM I
C
C     I              INDEX
C     IERROR         INDICATOR, 0 IF NO INPUT ERROR, 1 IF AN INPUT ERROR OCCURRED
C     J              INDEX
C     JJ             INDEX
C     K              INDEX
C     KK             INDEX
C     L(46)          TEMPORARY WORKING ARRAY
C     LD(3)          DATE OF RUN
C     LM(23)         SYSTEM MODE NAME
C     LN(23)         SYSTEM NAME
C     LS(8)          SYSTEM CODE
C     LT(8)          TEMPORARY WORKING ARRAY
C     M              INDEX
C     MARK           I.D. NO. IN COL. 80 OF INPUT CARDS
C     MCHECK(4)      SYSTEM CODE SAVED IN THIS ARRAY
C     MCODE          SYSTEM MODE NUMBER
C     MENT           NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
C     N              INDEX
C     NEL            NUMBER OF ELEMENTS IN SYSTEM
C     NIT(25)        NUMBER OF INTERMEDIATE TIMES FOR EACH TRANSITION INTERVAL
C     NN             INDEX
C     NST(50)        ENTRIES OF THE SYSTEM DEFINITION ARRAY
C     NT             NUMBER OF TRANSITION TIMES
C     TT(25)         TRANSITION TIMES
C
      SUBROUTINE RAP001 (LS,LN,LD,MCODE,LM,NEL,NST,TT,NIT,MENT,NT,IERROR
     1,MCHECK)
      DIMENSION LS(8),LN(23),LD(3),LM(23),NST(50),TT(25),NIT(25),LT(8),L
     1(46)
      DIMENSION MCHECK(4)
  100 FORMAT (3X,8A1,6X,23A2,6X,I2,1X,I2,1X,I2,I3)
  101 FORMAT (I3,14X,23A2,15X,I2)
  102 FORMAT (I3,17X,50A1,8X,I2)
  103 FORMAT (I3,5(1X,E10.2,1X,I3),I2)
  104 FORMAT (3X,5(1X,E10.2,1X,I3),I2)
  105 FORMAT (13H SYSTEM CODE ,8A1,17H WITH SYSTEM MODE,I2,20H PREVIOUSL
  1051Y ASSIGNED)
  106 FORMAT (1H ,I2,21H CARD OUT OF SEQUENCE)
  107 FORMAT (33H ERROR IN SYSTEM DEFINITION ARRAY)
  108 FORMAT(1H ,25HDATA SET IGNORED - RAP001)
C
C          IF ERROR, FIND BEGINNING OF NEXT DATA SET
C
      IF(IERROR-1)49,47,49
   47 I=FRSTRD(80)
      IF(I-71)56,55,56
   55 READ 100,(LS(I),I=1,8),(LN(I),I=1,23),(LD(I),I=1,3),MARK
      IERROR=0
      GO TO 1
   56 READ 101,I
      GO TO 47
C
C          READ TYPE 1 CARD (SYSTEM NAME)
C
   49 READ 100,(LS(I),I=1,8),(LN(I),I=1,23),(LD(I),I=1,3),MARK
      IF(MARK-1)9,1,9
C
```

Figure B-49. RAP001 Subprogram, Program Listing (1 of 3)

```
C          READ TYPE 2 CARD (SYSTEM MODE NAME)
C
      1 READ 101,MCODE,(LM(I),I=1,23),MARK
        IF(MARK-2)9,26,9
C
C          CHECK SYSTEM AND MODE CODE
C
     26 DO 2 I=1,8
      2 LT(I)=0
        J=1
        DO 4 I=1,8
        IF(LS(I))3,4,3
      3 LT(J)=LS(I)
        J=J+1
      4 CONTINUE
        LT(1)=LT(1)+LT(2)/100
        LT(2)=LT(3)+LT(4)/100
        LT(3)=LT(5)+LT(6)/100
        LT(4)=LT(7)+LT(8)/100
        LT(5)=MCODE
        DO 57 J=1,4
     57 MCHECK(J)=LT(J)
        IF(MENT)6,10,6
      6 DO 8 I=1,MENT
        J=2050+I
        FETCH (J)(NST(K),K=1,5)
        DO 7 J=1,5
        IF(LT(J)-NST(J))8,7,8
      7 CONTINUE
C
C          ERROR IN SYSTEM AND MODE CODE
C
        PRINT 105,LS,MCODE
        GO TO 32
      8 CONTINUE
        GO TO 10
C
C          CARD SEQUENCE ERROR
C
      9 PRINT 106,MARK
        GO TO 32
C
C          CENTER SYSTEM NAME
C
     10 DO 11 I=1,23
        L(2*I-1)=LN(I)/100
     11 L(2*I)=LN(I)-(LN(I)/100)*100
        KK=1
        GO TO 28
     30 DO 21 I=1,23
     21 LN(I)=L(2*I-1)*100+L(2*I)
C
C          CENTER MODE NAME
C
     25 DO 27 I=1,23
        L(2*I-1)=LM(I)/100
     27 L(2*I)=LM(I)-(LM(I)/100)*100
        KK=2
        GO TO 28
     31 DO 37 I=1,23
     37 LM(I)=L(2*I-1)*100+L(2*I)
        GO TO 41
C
C          CENTERING ROUTINE
C
```

Figure B-49. RAP001 Subprogram, Program Listing (2 of 3)

```
28 N=0
   DO 13 I=1,46
   IF(L(I))14,12,14
12 N=N+1
13 CONTINUE
   GO TO 33
14 M=0
   J=46
   DO 16 I=1,46
   IF(L(J))17,15,17
15 M=M+1
16 J=J-1
17 NN=(N+M)/2
   IF(N-NN)18,33,22
18 K=NN-N
   NN=46-N-M
   DO 19 I=1,NN
   J=47-M-I
   JJ=J+K
19 L(JJ)=L(J)
   DO 20 I=1,K
   J=JJ-I
20 L(J)=0
   GO TO 29
22 K=N-NN
   NN=46-N-M
   DO 23 I=1,NN
   J=N+I-1
   JJ=J-K
23 L(JJ)=L(J)
   DO 24 I=1,K
   J=JJ+I
24 L(J)=0
29 GO TO (30,31),KK
33 GO TO (25,41),KK
C
C          READ TYPE 3 CARD (SYSTEM DEFINITION ARRAY)
C
41 READ 102,NEL,(NST(I),I=1,50),MARK
   IF(MARK-3)9,42,9
C
C          CHECK SYSTEM DEFINITION ARRAY ENTRIES
C
42 NN=NEL+1
   DO 52 I=1,NEL
   IF(NST(I))52,51,52
52 NST(I)=NST(I)/100-70
   IF (NEL-50)53,54,54
53 DO 50 I=NN,50
   IF(NST(I))51,50,51
50 CONTINUE
C
C          READ TYPE 4 CARDS (TIMES)
C
54 READ 103,NT,(TT(I),NIT(I),I=1,5),MARK
   IF(MARK-4)9,43,9
43 IF(NT-5)46,46,44
44 DO 45 I=6,NT,5
   READ 104,TT(I),NIT(I),TT(I+1),NIT(I+1),TT(I+2),NIT(I+2),TT(I+3),NI
  1T(I+3),TT(I+4),NIT(I+4),MARK
   IF(MARK-4)9,45,9
45 CONTINUE
46 RETURN
C
C          ERROR IN SYSTEM DEFINITION ARRAY ENTRY
C
51 PRINT 107
32 PRINT 108
   GO TO 47
   END
```

Figure B-49. RAP001 Subprogram, Program Listing (3 of 3)

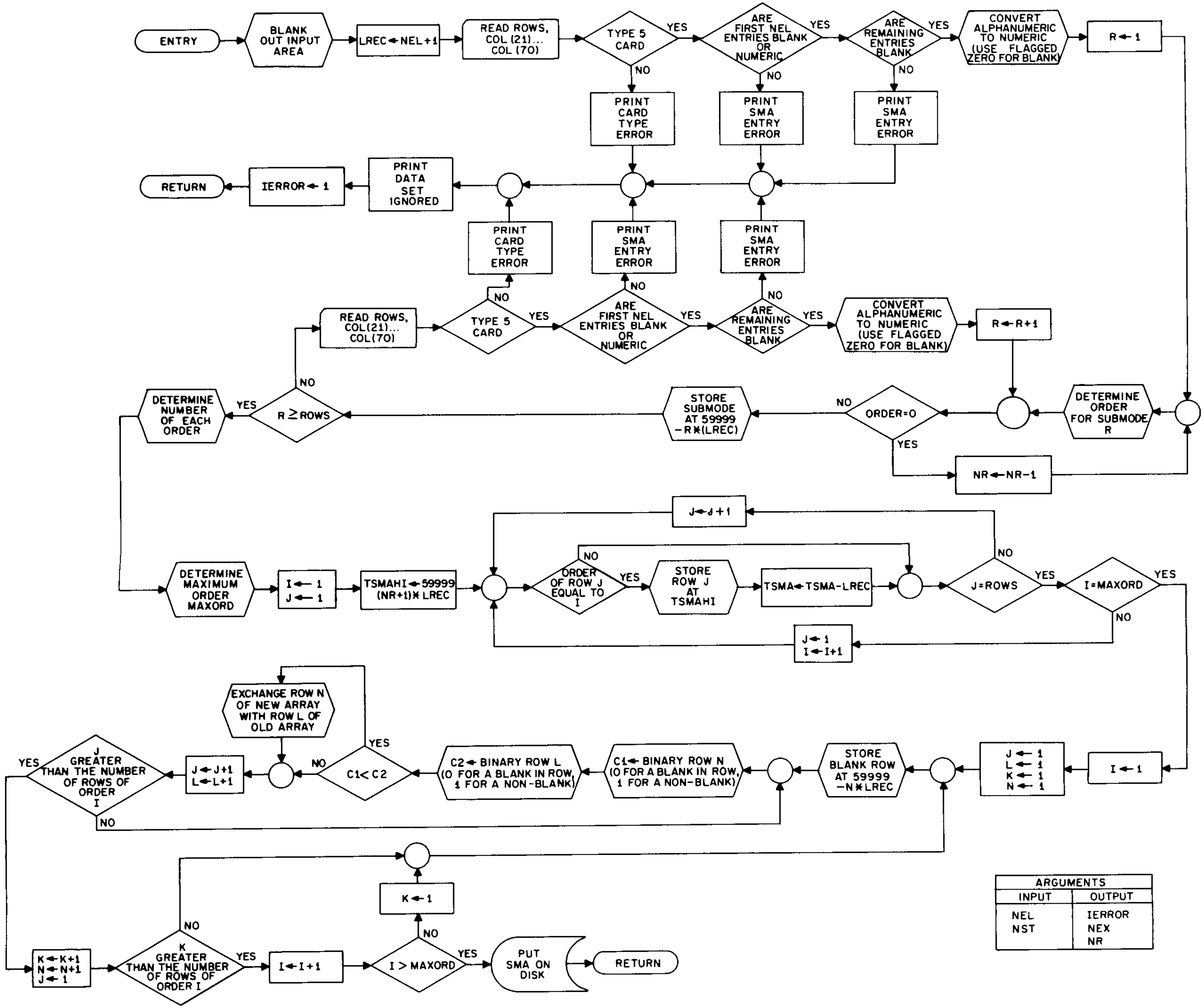


Figure B-50. RAP002 Subprogram, Logic Flow Chart

```
*
*     RAP002 - - - INPUT FROM DATA TRANSMITTAL FORM II
*     AND REARRANGEMENT OF THE SYSTEM MODE ARRAY BY ORDER
*
*     SUBROUTINE LINKAGE
*
S      DS   ,*+101
       DC   6,987898,5-S
       DAC  6,RAP002,7-S
       DVLC22-S,5,LAST,2,08,2,04,5,RAP002-6,5,0,30,0
       DSC  17,0,0
       DORGS-100
INSUB  DSA  0
NST    DSA  0
NEX    DSA  0
NROW   DSA  0
IERRORDSA   0
NEL    DS   0,INSUB
       DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*      C1           FIELD TO CONTAIN BINARY REPRESENTATION
*      C2           FIELD TO CONTAIN BINARY REPRESENTATION
*      COUNT1       COUNTER
*      COUNT2       COUNTER
*      I            INDEX
*      IERROR       INDICATOR, 0 IF NO INPUT ERROR, 1 IF AN INPUT ERROR
*                   OCCURRED
*      INPUT        INPUT AREA
*      J            INDEX
*      K            INDEX
*      LREC         LENGTH OF RECORD FOR EACH ROW OF THE SYSTEM MODE ARRAY
*      MAXORD       LARGEST ORDER PRESENT IN CURRENT SYSTEM MODE ARRAY
*      NEL          NUMBER OF ELEMENTS IN SYSTEM
*      NEX          NUMBER OF SPACING LINES TO APPEAR ON INTERMEDIATE
*                   PRINT OUTS
*      NROW         NUMBER OF ROWS IN SYSTEM MODE ARRAY
*      NST(50)      ENTRIES OF THE SYSTEM DEFINITION ARRAY
*      NUMOR(50)    ARRAY CONTAINING THE NUMBER OF EACH ORDER
*      ORDER(100)   ARRAY CONTAINING THE ORDER OF EACH ROW IN THE SYSTEM
*                   MODE ARRAY
*      R            INDEX
*      RESET        ADDRESS INITIALIZATION CONSTANT
*      RMK          RECORD MARK
*      ROWS         NUMBER OF ROWS IN SYSTEM MODE ARRAY
*      SAVES        ARRAY ADDRESS INDEX
*      SAVET        ARRAY ADDRESS INDEX
*      SMAHI        ARRAY ADDRESS INDEX
*      TOP          HIGHEST ADDRESS IN MEMORY
*      TSMAHI       ARRAY ADDRESS INDEX
*      WORK         WORK AREA
*      ZERO         CONSTANT TO INITIALIZE WORK AREA
*
INPUT  DAS  81
RESET  DSA  0
SMAHI  DSA  0
TSMAHIDSA   0
SAVES  DSA  0
SAVET  DSA  0
```

Figure B-51. RAP002 Subprogram, Program Listing (1 of 8)

```
LREC   DC   4,0
       DS   1
NUMOR  DC   2,00
       DC   48,000000000000000000000000000000000000000000000000
       DC   50,00000000000000000000000000000000000000000000000000
ORDER  DC   2,00
       DC   48,000000000000000000000000000000000000000000000000
       DC   50,00000000000000000000000000000000000000000000000000
       DC   50,00000000000000000000000000000000000000000000000000
       DC   50,00000000000000000000000000000000000000000000000000
ROWS   DS   3
R      DS   3
I      DS   3
J      DS   3
COUNT1DS    2
COUNT2DS    2
MAXORDDS    2
TOP    DS   ,59999
C1     DSS  50
       DC   1,'
C2     DSS  50
       DC   1,'
ERMES  DAC  21,TYPE 5 CARD MISSING,'
ERMES1DAC   32,ENTRY FOR ELEMENT NOT IN SYSTEM'
ERMES2DAC   21,DATA WILL BE IGNORED'
ERMES3DAC   37,BLANK ROW IN SMA,EXECUTION CONTINUES'
ERMES4DAC   25,ILLEGAL CHARACTER IN SMA'
K      DS   3
       DS   1
WORK   DSS  50
RMK    DC   1,'
ZERO   DSC  50,00000000000000000000000000000000000000000000000000
       DC   1,'
MES2   DAC  50,ELEMENT MODE NUMBER IN SMA IS LARGER THAN THAT SPE
       DAC  15,CIFIED IN SDA,'
DSKOUTDDA   ,1,10100,1,WORK
       DC   1,'
OUTGO  DD   ,DSKOUT,,,A
D2     DDA  ,1,10000,1,NUMOR-1
       DC   1,'
DORDERDD    ,D2,,,A
CARD   DCA  ,INPUT
ER     DPRA,MES2
ERR    DPRA,ERMES
ERR1   DPRA,ERMES1
ERR2   DPRA,ERMES2
ERR3   DPRA,ERMES3
ERR4   DPRA,ERMES4
*
*      TRANSFER OF ARGUMENTS
*
       DC   5,0
RAP002TFM   TF+6,INSUB-4
       AM   TF+6,4,10
       AM   RAP002-1,5,10
       TF   CF+11,RAP002-1,11
       BNF  *+36,CF+11
CF     CF   CF+11
       TF   CF+11,CF+11,11
TF     TF   INSUB,CF+11
```

Figure B-51. RAP002 Subprogram, Program Listing (2 of 8)

```
        AM   TF+6,1,10
        BNR  RAP002+12,TF+6,11
        AM   RAP002-1,2,10
*
*     START OF SUBROUTINE
*
INDEXIBBASBANDA
        BBBSBANDB
        TDM  XON      +11,0,,    SET BAND 0 EXIT
        B7   BANDO
BANDA   TDM  XON      +11,1,,    SET BAND 1 EXIT
        B7   BANDO
BANDB   TDM  XON      +11,2,,    SET BAND 2 EXIT
BANDO   BS   *+12,0,,            SELECT BAND 0 NOW
        TFM  *+18,INPUT-1
CLEAR   TDM  INPUT,0
        CM   CLEAR+6,INPUT+160
        BE   *+32
        AM   CLEAR+6,1,10
        B7   CLEAR
        TFM  LOOP3+43,1,10
        TF   LREC,NEL,11
        AM   LREC,1,10
        TFM  RESET,TOP
        S    RESET,LREC
        TFM  *+30,ZERO
        A    *+18,NEL,11
REMOVETD     0,RMK
*
*     READ FIRST TYPE 5 CARD (NUMBER OF ROWS AND FIRST ROW)
*
        GET  CARD
        BTM  TYPE5,*+12
        BTM  CHECK,*+12
        BTM  ALPHA,*+12
        TD   ROWS-2,INPUT
        TD   ROWS-1,INPUT+2
        TD   ROWS,INPUT+4
        SF   ROWS-2
*
*     DETERMINE SUBMODE ORDER
*
        TFM  REF+6,ORDER
        TF   SMAHI,RESET
        TFM  COUNT1,0,10
        TFM  R,1,9
        TFM  LOOP1+11,WORK
LOOP1   BNF  *+20,WORK
        B7   *+20
        AM   COUNT1,1,10
        AM   LOOP1+11,1,10
        BNR  LOOP1,LOOP1+11,11
REF     TF   ORDER,COUNT1
        CM   *-6,0,610
        BE   NULL
        TFM  COUNT1,0,10
        TR   SMAHI,WORK,6
        S    SMAHI,LREC
        AM   REF+6,2,10
        C    R,ROWS
```

Figure B-51. RAP002 Subprogram, Program Listing (3 of 8)

```
        BE   LOOP2-48
        AM   R,1,10
*
*     READ ADDITIONAL TYPE 5 CARDS
*
ADD     GET  CARD
        BTM  TYPE5,*+12
        BTM  CHECK,*+12
        BTM  ALPHA,*+12
        B7   LOOP1-12
*
*     DETERMINE NUMBER OF EACH ORDER
*
        TFM  LOOP2+18,ORDER
        TF   TSMAHI,SMAHI
        TF   SMAHI,RESET
        TFM  R,0,9
LOOP2   AM   R,1,10
        MM   ORDER,2,10
        SF   00095
        AM   00099,NUMOR-2
        TF   *+18,00099
        AM   NUMOR-2,1,10
        AM   LOOP2+18,2,10
        C    R,ROWS
        BE   *+20
        B7   LOOP2
*
*     DETERMINE THE MAXIMUM ORDER AND THE NUMBER OF BLANK
*     LINES AFTER EACH ORDER ON THE OUTPUT PAGES
*
        TFM  NEX,1,6811
        TFM  LOOP3+6,NUMOR
LOOP3   CM   NUMOR,0,10
        BNE  *+20
        B7   *+32
        TFM  MAXORD,1,10
        AM   NEX,1,610
        AM   *-13,1,10
        CM   LOOP3+6,NUMOR+98
        BE   *+36
        AM   LOOP3+6,2,10
        B    LOOP3
*
*     REARRANGE SMA BY ORDER (LOWEST ORDER FIRST)
*
        TFM  L4+6,ORDER
        TFM  I,1,9
        TFM  J,1,9
L4      C    ORDER,I
        BE   L5
        AM   *-18,2,10
        C    J,ROWS
        BE   L5+32
        AM   J,1,10
        S    SMAHI,LREC
        B7   L4
L5      TR   TSMAHI,SMAHI,611
        S    TSMAHI,LREC
        B7   L4+24
```

Figure B-51. RAP002 Subprogram, Program Listing (4 of 8)

```
         C     I,MAXORD
         BL    NEXT
         AM    I,1,10
         TF    SMAHI,RESET
         TFM   L4+6,ORDER
         B7    L4-12
*
*      REARRANGE SMA WITHIN EACH GROUP OF ORDERS
*
NEXT     TFM   I,0,9
         TFM   REF1+11,NUMOR-2
         TFM   REF1+59,NUMOR-2
         TF    SAVET,SMAHI
         S     SAVET,LREC
         TF    SAVES,RESET
         AM    I,1,10
         AM    REF1+11,2,10
         AM    REF1+59,2,10
L1       TFM   K,1,9
         TF    SMAHI,SAVES
         TFM   J,1,9
         TF    TSMAHI,SAVET
         CM    REF1+11,0,610
         BE    REF1+72
         TR    SMAHI,ZERO,6
L2       TR    C1,SMAHI,11
         TR    C2,TSMAHI,11
         BNF   *+32,C1
         TDM   L2+35,0,6
         B7    *+20
         TDM   L2+35,1,6
         AM    L2+35,1,10
         BNR   L2+24,L2+35,11
MASK     BNF   *+32,C2
         TDM   MASK+11,0,6
         B7    *+20
         TDM   MASK+11,1,6
         AM    MASK+11,1,10
         BNR   MASK,MASK+11,11
         TFM   L2+35,C1
         TFM   MASK+11,C2
         SF    C1
         SF    C2
         TFM   *+54,C1-1
         TFM   *+47,C2-1
         A     *+30,NEL,11
         A     *+23,NEL,11
         C     C1,C2
         BNL   *+48
         TR    WORK,SMAHI,11
         TR    SMAHI,TSMAHI,611
         TR    TSMAHI,WORK,6
         AM    J,1,10
         S     TSMAHI,LREC
REF1     C     J,NUMOR-2
         BNH   L2
         AM    K,1,10
         S     SMAHI,LREC
         C     K,NUMOR-2
         BNH   L1+24
```

Figure B-51. RAP002 Subprogram, Program Listing (5 of 8)

```
         C     I,MAXORD
         BE    STORE
         TF    SAVES,SMAHI
         TF    SAVET,TSMAHI
         B7    NEXT+72
*
*        RESTORE ZEROS AND RETURN
*
XON      BS    RAP002-1,,6
OUT      TFM   I,1,9
         TFM   *+18,ROWS-3
         TFM   0,0,10
         CM    I,150,9
         BE    XON
         SM    OUT+30,2,10
         AM    I,1,10
         B     OUT+24
*
*        ALPHAMERIC TO NUMERIC INPUTS
*
ALPHA    TFM   ALPHA+83,INPUT+40
         TFM   ALPHA+95,INPUT+39
         TFM   EMPTY-14,WORK
         TFM   EMPTY+23,INPUT+38
         A     EMPTY+23,NEL,11
         A     EMPTY+23,NEL,11
         TD    COUNT2,INPUT+40
         TD    COUNT2-1,INPUT+39
         SF    COUNT2-1
         CM    COUNT2,70,10
         BL    EMPTY
         TD    WORK,ALPHA+83,11
         B7    *+20
EMPTY    TDM   EMPTY-14,,611
         CM    ALPHA+83,INPUT+38
         BE    *+56
         AM    EMPTY-14,1,10
         AM    ALPHA+83,2,10
         AM    ALPHA+95,2,10
         B7    ALPHA+72
         TF    *+30,EMPTY-14
         AM    *+18,1,10
         TD    WORK,RMK
         B     ALPHA-1,,6
*
*        TEST FOR ALLOWED ENTRY IN SMA
*
CHECK    TDM   COUNT2,0
         TFM   LOOK+11,INPUT+39
         TFM   LOOK+95,INPUT+40
         TF    LOOK+119,NST
         TF    LOOK+143,NEL
         TFM   K,0,9
         AM    K,1,10
LOOK     TD    COUNT2-1,INPUT+39
         SF    COUNT2-1
         CM    COUNT2,70,10
         BE    *+48
         CM    COUNT2,0,10
         BNE   ERROR3
```

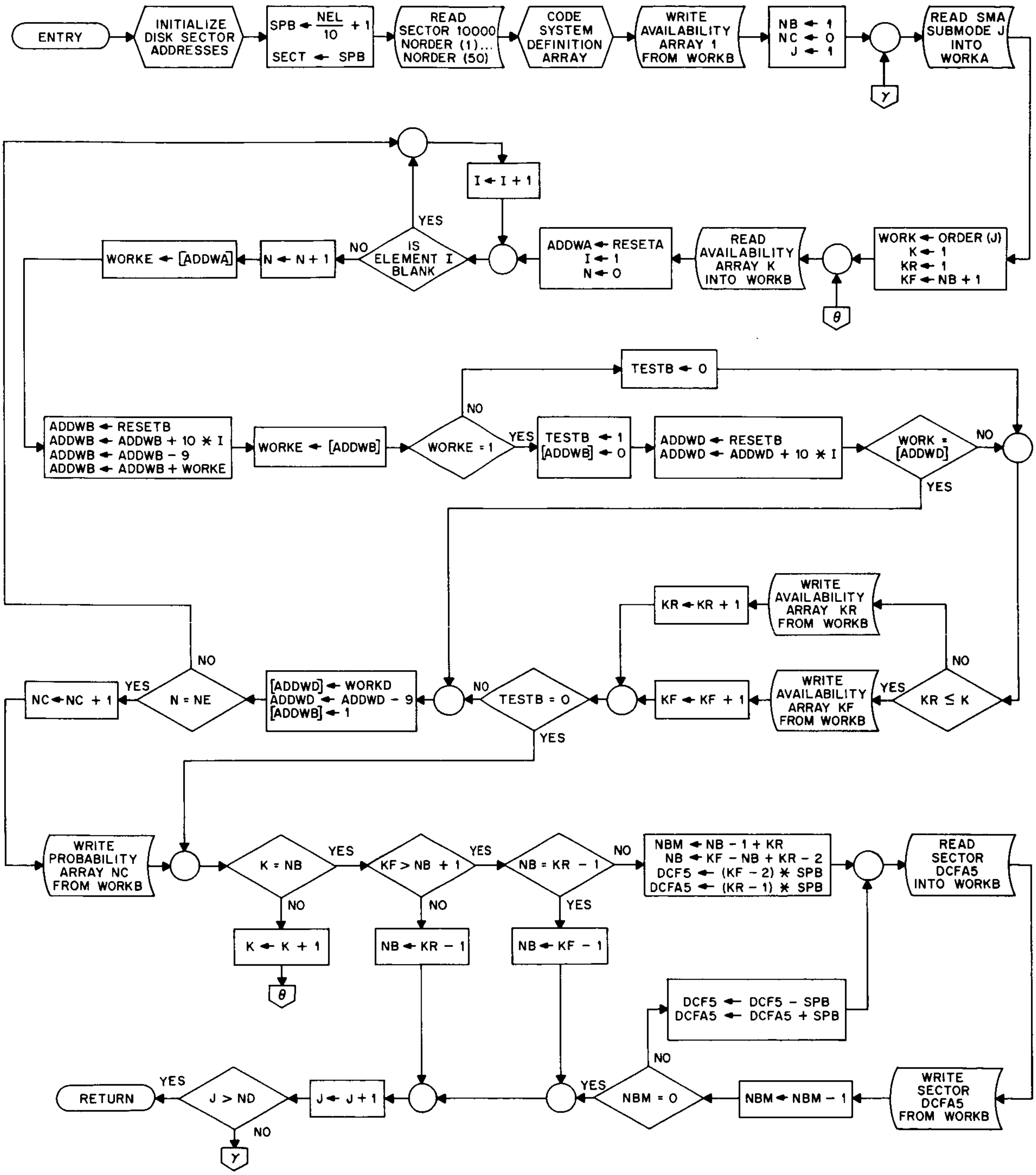
Figure B-51. RAP002 Subprogram, Program Listing (6 of 8)

```
        B    NEXONE-24
        TD   COUNT2,INPUT+40
        TDM  COUNT2-1,0,11
        C    COUNT2,NST
        BH   ERROR2
        C    K,NEL
        BE   OTHER
NEXONEAM     LOOK+119,4,10
        AM   LOOK+95,2,10
        AM   LOOK+11,2,10
        TDM  COUNT2,0
        B    LOOK-12
OTHER   TF   HUNT+11,LOOK+11
        TF   HUNT+23,LOOK+95
        AM   K,1,10
        CM   K,50,9
        BH   CHECK-1,,6
        AM   HUNT+11,2,10
        AM   HUNT+23,2,10
HUNT    TD   COUNT2-1,0
        TD   COUNT2,0
        SF   COUNT2-1
        CM   COUNT2,0,10
        BE   *+20
        B7   ERROR1
        B    OTHER+24
*
*      TEST FOR TYPE 5 CARD
*
TYPE5   SF   INPUT+157
        CM   INPUT+158,75,10
        BNE  ERROR
        CF   INPUT+157
        B    TYPE5-1,,6
*
*      STORE SYSTEM MODE ARRAY ON DISK
*
STORE   TFM  COUNT1,0,10
        TFM  *+71,TOP
        TFM  DSKOUT+5,10099
        AM   COUNT1,1,10
        S    *+35,LREC
        AM   DSKOUT+5,1,10
        TR   WORK,TOP
        PUT  OUTGO
        C    COUNT1,ROWS
        BE   *+20
        B7   STORE+36
        TDM  REMOVE+6,0,611
        TF   NROW,ROWS,6
        PUT  DORDER
        B    OUT
*
*      BLANK ROW ENTERED IN SYSTEM MODE ARRAY
*
NULL    SM   ROWS,1,10
        SM   R,1,10
        TFM  COUNT1,0,10
        RCTY
        PUT  ERR3
```

Figure B-51. RAP002 Subprogram, Program Listing (7 of 8)

```
        C    R,ROWS
        BE   LOOP2-36
        B7   ADD
*
*     ILLEGAL ENTRY IN SYSTEM MODE ARRAY
*
ERROR3RCTY
        PUT ERR4
        RCTY
        PUT ERR2
        TFM IERROR,1,68
        B7   OUT
*
*     ERROR - - ENTRY FOR ELEMENT NOT IN SYSTEM
*
ERROR1RCTY
        PUT ERR1
        RCTY
        PUT ERR2
        TFM IERROR,1,68
        B7   OUT
*
*     CARD SEQUENCE ERROR
*
ERROR RCTY
        PUT ERR
        RCTY
        PUT ERR2
        TFM IERROR,1,68
        B7   OUT
*
*     ERROR IN SMA ENTRY
*
ERROR2RCTY
        PUT ER
        RCTY
        PUT ERR2
        TFM IERROR,1,68
        B7   OUT
LAST    DC   2, '
        DENDRAP002
```

Figure B-51. RAP002 Subprogram, Program Listing (8 of 8)



ARGUMENTS	
INPUT	OUTPUT
MCODE	NARRAY
NEL	SECT
NR	
NST	

Figure B-52. RAP003 Subprogram, Logic Flow Chart

```
*
*     RAP003 - - - LOGICAL MANIPULATION OF SYSTEM MODE ARRAY
*
*     SUBROUTINE LINKAGE
*
S       DS   ,*+101
        DC   6,987898,5-S
        DAC  6,RAP003,7-S
        DVLC22-S,5,LAST,2,08,2,04,5,RAP003-6,5,0,30,0
        DSC  17,0,0
        DORGS-100
NEL     DSA  0
NST     DSA  0
NROWS   DSA  0
NARRAYDSA  0
SECT    DSA  0
MCODE   DSA  0
        DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*       ADDWA        ARRAY ADDRESS INDEX
*       ADDWB        ARRAY ADDRESS INDEX
*       ADDWB        ARRAY ADDRESS INDEX
*       DCF5         DISK SECTOR ADDRESS
*       DCFA5        DISK SECTOR ADDRESS
*       I            INDEX
*       J            INDEX
*       K            INDEX
*       KF           INDEX
*       KR           INDEX
*       KRA          INDEX
*       MCODE        SYSTEM MODE NUMBER
*       N            INDEX
*       NARRAY       NUMBER OF PROBABILITY ARRAYS
*       NB           NUMBER OF AVAILABILITY ARRAYS
*       NBA          INDEX
*       NBM          INDEX
*       NC           NUMBER OF PROBABILITY ARRAYS
*       NEL          NUMBER OF ELEMENTS IN SYSTEM
*       NORDER(50)   ARRAY CONTAINING THE NUMBER OF EACH ORDER
*       NROWS        NUMBER OF ROWS IN SYSTEM MODE ARRAY
*       NST(50)      ENTRIES OF THE SYSTEM DEFINITION ARRAY
*       ORD          INDEX
*       ORDER        ORDER OF CURRENT SMA SUBMODE
*       RESETA       ADDRESS INITIALIZATION CONSTANT
*       RESETB       ADDRESS INITIALIZATION CONSTANT
*       RMARK        RECORD MARK
*       SECT         NUMBER OF SECTORS PER ARRAY
*       SPB          NUMBER OF SECTORS PER ARRAY
*       TEMP         WORKING VARIABLE
*       TESTB        SWITCH
*       WORKA        WORKING ARRAY
*       WORKB        WORKING ARRAY
*       WORKD        WORKING ARRAY
*       WORKE        WORKING VARIABLE
*       WORKF        WORKING VARIABLE
*
TEMP    DS   4
        DS   1
```

Figure B-53. RAP003 Subprogram, Program Listing (1 of 5)

```
WORKB DSS 601
      DS  1
WORKA DSS 100
NORDERDAS 50
WORKD DC  10,0
WORKE DC  2,0
WORKF DC  2,0
ADDWA DSA WORKA-1
ADDWB DSA WORKB-1
ADDWD DSA WORKB-1
RESETADSA WORKA-1
RESETBDSA WORKB-1
RMARK DC  1,'
DCF   DDA ,5,00000,001,WORKB
      DC  1,'
DCF5  DS  ,DCF+5
DEFD  DD  ,DCF,,,A
DCFP  DDA ,3,19800,001,WORKB
      DC  1,'
DEFDP DD  ,DCFP,,,A
DCFA  DDA ,5,00000,001,WORKB
      DC  1,'
DCFA5 DS  ,DCFA+5
DEFDA DD  ,DCFA,,,A
DKSMA DDA ,1,10100,1,WORKA
      DC  1,'
DSMA  DD  ,DKSMA,,,A
D1    DDA ,1,10000,1,NORDER-1
      DC  1,'
DORDERDD  ,D1,,,A
NB    DC  5,1
NC    DC  5,1
N     DC  2,1
I     DS  3
J     DS  3
ORD   DS  2
ORDER DS  2
K     DC  5,1
KR    DC  5,1
KF    DC  5,1
KRA   DC  5,1
NBM   DC  5,1
NBA   DC  5,1
TESTB DC  2,1
SPB   DC  3,1
      DC  5,0
*
*     TRANSFER OF ARGUMENTS
*
RAP003TFM TF+6,NEL-4
      AM  TF+6,4,10
      AM  RAP003-1,5,10
      TF  CF+11,RAP003-1,11
      BNF *+36,CF+11
CF    CF  CF+11
      TF  CF+11,CF+11,11
TF    TF  NEL,CF+11
      AM  TF+6,1,10
      BNR RAP003+12,TF+6,11
      AM  RAP003-1,1,10
```

Figure B-53. RAP003 Subprogram, Program Listing (2 of 5)

```
*
*       START OF SUBROUTINE
*
          CM   MCODE,0,68
          BE   *+32
          TFM  DKSMA+5,10099
          B7   *+20
          TFM  DKSMA+5,10199
          TFM  XC+6,WORKB+3
          A    XC+5,NEL,11
          TFM  XD+6,WORKB+2
          A    XD+5,NEL,11
          TF   TEMP,NEL,11
          TD   SPB,TEMP-1
          AM   SPB,1,10
          TF   DCF+8,SPB
          TF   DCFP+8,SPB
          TF   DCFA+8,SPB
          TF   SECT,SPB,6
          GET  DORDER
*
*       CODE SDA AND SEND TO AVAILABLE AREA ON DISK
*
          TFM  I,1,10
          TFM  REF+42,WORKB
          TF   *+23,NST
          TF   TEMP,NST
REF       TFM  J,0,10
          C    J,TEMP
          BNH  *+32
          TDM  REF+42,0,6
          B7   *+20
          TDM  REF+42,1,6
          AM   REF+42,1,10
          CM   J,9,10
          BNL  *+32
          AM   J,1,10
          B7   REF+12
          C    I,NEL,11
          BL   *+20
          B7   *+40
          AM   I,1,10
          AM   REF-1,4,10
          B7   REF-12
          TFM  DCFP+5,19800
XC        TD   WORKB+3,RMARK
          TFM  DCF+5,0
          PUT  DEFD
*
*       STRIP ROUTINE
*
          TFM  NB,1
          TFM  NC,0,8
          TFM  J,1,9
          TFM  C+11,NORDER
          SM   C+11,2,10
          TFM  ORDER,0,10
          TFM  ORD,1,10
          AM   ORDER,1,10
          AM   *+23,2,10
```

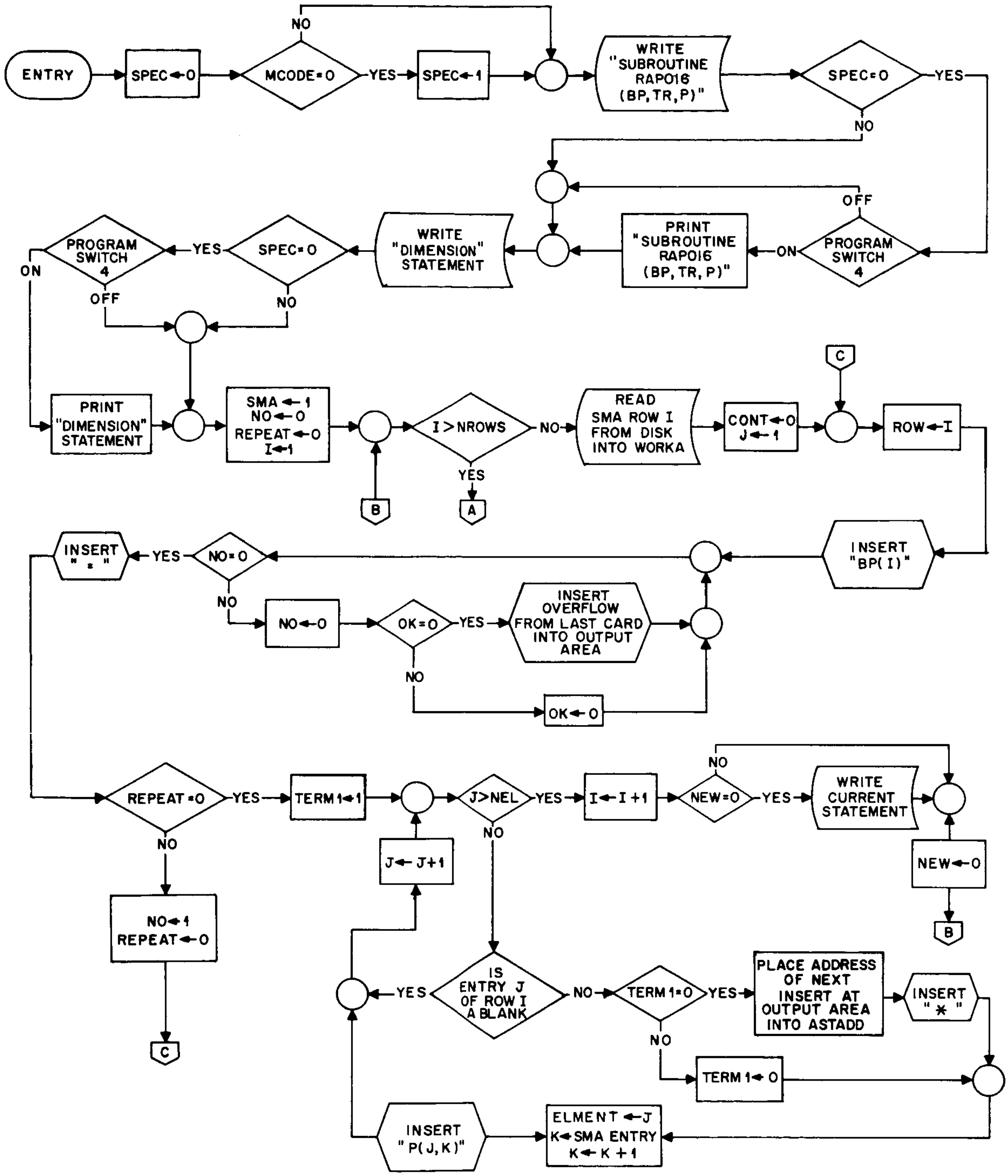
Figure B-53. RAP003 Subprogram, Program Listing (3 of 5)

```
C       C    ORD,NORDER
        BH   C-36
        AM   DKSMA+5,1,10
        GET  DSMA
        AM   ORD,1,10
        TFM  K,1
        TFM  KR,1
        TF   KF,NB
        AM   KF,1,10
B       TF   ADDWA,RESETA
        SM   K,1,10
        M    K,SPB
        SF   95
        AM   K,1,10
        TF   DCF+5,99
        GET  DEFD
        TFM  I,1,10
        TFM  N,0,10
L1      AM   ADDWA,1,10
        BNF  TEA,ADDWA,11
L1A     AM   I,1,10
        B7   L1
TEA     TD   WORKE,ADDWA,11
        AM   N,1,10
        TF   ADDWB,RESETB
        A    ADDWB-1,I
        SM   ADDWB,9,10
        A    ADDWB,WORKE
        TD   WORKE,ADDWB,11
        CM   WORKE,1,10
        BE   L5
        TDM  TESTB,0
L2      C    KR,K
        BH   L3
        SM   KR,1,10
        M    KR,SPB
        SF   95
        TF   DCF+5,99
        PUT  DEFD
        AM   KR,2,10
        B7   L4
L3      SM   KF,1,10
        M    KF,SPB
        SF   95
        TF   DCF+5,99
        PUT  DEFD
        AM   KF,2,10
L4      CM   TESTB,0
        BNE  L6
        B7   A
L5      TDM  TESTB,1
        TDM  ADDWB,0,6
        TF   ADDWD,RESETB
        A    ADDWD-1,I
        C    WORKD,ADDWD,11
        BNE  L2
L6      TF   ADDWD,WORKD,6
        SM   ADDWD,9,10
        CF   -ADDWD
        TDM  ADDWB,1,6
```

Figure B-53. RAP003 Subprogram, Program Listing (4 of 5)

```
         C    N,ORDER
         BNE  L1A
XD       TF   WORKB+2,J
         S    DCFP+5,SPB
         PUT  DEFDP
         AM   NC,1,10
A        C    NB,K
         BE   L7
         AM   K,1,10
         B7   B
L7       TF   NBA,NB
         AM   NBA,1,10
         C    KF,NBA
         BH   L9
         TF   NB,KR
         SM   NB,1,10
L8       AM   J,1,10
         C    J,NROWS,11
         BNH  C
         TF   NARRAY,NC,6
         B7   RAP003-1,,6
L9       TF   KRA,KR
         SM   KRA,1,10
         C    NB,KRA
         BNE  L10
         TF   NB,KF
         SM   NB,1,10
         B7   L8
L10      TF   NBM,NB
         AM   NBM,1,10
         S    NBM,KR
         SF   NB
         A    NB,KF
         A    NB,KR
         SM   NB,2,10
         SM   KF,2,10
         M    KF,SPB
         SF   95
         TF   DCF5,99
         SM   KR,1,10
         M    KR,SPB
         SF   95
         TF   DCFA5,99
L11      GET  DEFD
         PUT  DEFDA
         SM   NBM,1,10
         BZ   L8
         S    DCF5,SPB
         A    DCFA5,SPB
         B7   L11
LAST     DC   2,0'
         DENDRAP003
```

Figure B-53. RAP003 Subprogram, Program Listing (5 of 5)



ARGUMENTS	
INPUT	OUTPUT
MCODE	
NARRAY	
NEL	
NR	
NST	
SECT	

Figure B-54. RAP004 Subprogram, Logic Flow Chart (1 of 3)

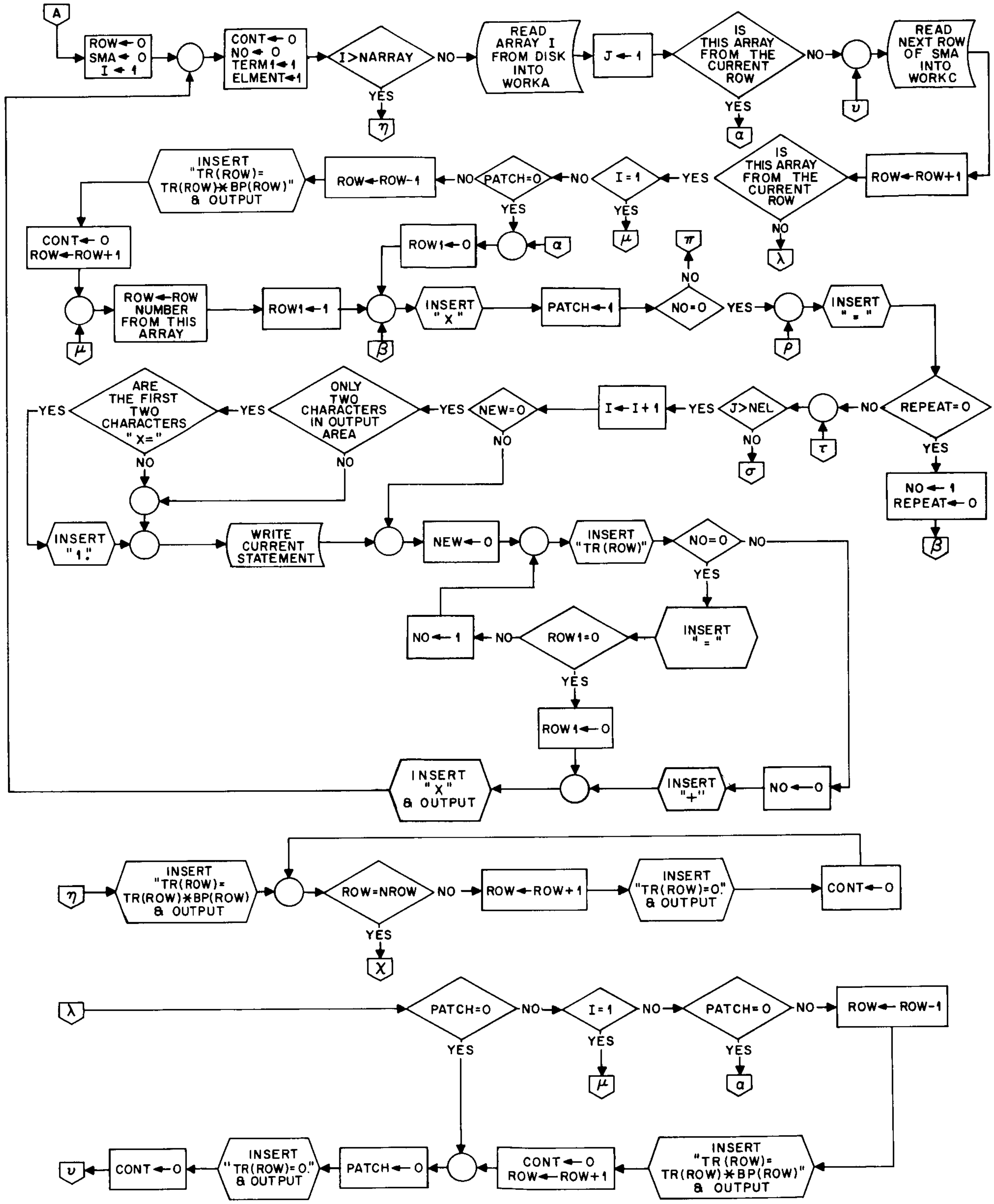


Figure B-54. RAP004 Subprogram, Logic Flow Chart (2 of 3)

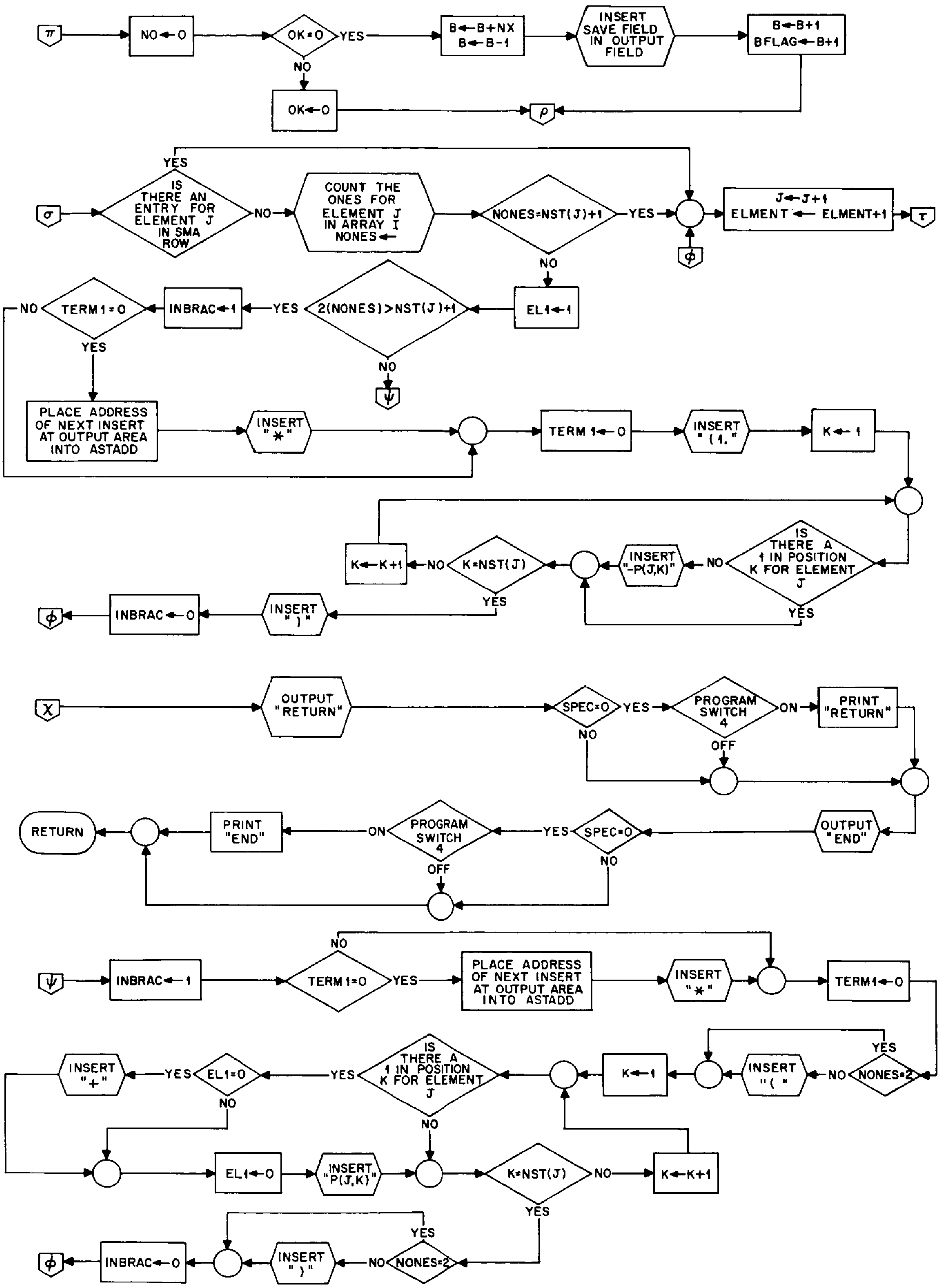


Figure B-54. RAP004 Subprogram, Logic Flow Chart (3 of 3)

```
*
*     RAP004 - - - GENERATION OF PROBABILITY EQUATION
*
*     SUBROUTINE LINKAGE
*
S       DS   ,*+101
        DC   6,987898,5-S
        DAC  6,RAP004,7-S
        DVLC22-S,5,BOTTOM,2,08,2,04,5,RAP004-6,5,0,30,0
        DSC  17,0,0
        DORGS-100
NEL     DSA  0
NST     DSA  0
NARRAYDSA  0
SECT    DSA  0
NROWS   DSA  0
MCODE   DSA  0
        DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*       ASTADD       ADDRESS OF LAST ASTRISK
*       B            OUTPUT ARRAY INDEX
*       BFLAG        OUTPUT ARRAY INDEX
*       CONT         CONTINUATION CARD NUMBER
*       EL1          SWITCH
*       ELMENT       ELEMENT NUMBER
*       I            INDEX
*       INBRAC       SWITCH
*       J            INDEX
*       K            INDEX
*       MCODE        SYSTEM MODE NUMBER
*       NARRAY       NUMBER OF PROBABILITY ARRAYS
*       NEL          NUMBER OF ELEMENTS IN SYSTEM
*       NEW          SWITCH
*       NO           SWITCH
*       NONES        NUMBER OF ELEMENT MODES IN TERM
*       NROWS        NUMBER OF ROWS IN SYSTEM MODE ARRAY
*       NST(50)      ENTRIES OF THE SYSTEM DEFINITION ARRAY
*       NX           LENGTH OF CARD OVERFLOW
*       OK           SWITCH
*       PATCH        SWITCH
*       REPEAT       SWITCH
*       RESET        OUTPUT AREA INITIALIZATION CONSTANT
*       ROW          CURRENT SMA ROW NUMBER
*       ROW1         SWITCH
*       SPEC         SWITCH
*       SECT         NUMBER OF SECTORS PER ARRAY
*       SMA          SIWTCH
*       TERM1        SIWTCH
*       WORKA        WORK AREA
*       WORKB        WORK AREA
*       WORKC        WORK AREA
*
PATCH DS   1
INBRACDS   1
REPEATDS , 1
SMA     DS   1
OK      DS   1
TERM1 DS   1
```

Figure B-55. RAP004 Subprogram, Program Listing (1 of 12)

```
ROW1   DS   1
NEW    DS   1
EL1    DS   1
NO     DS   1
CONT   DS   2
NONES  DS   2
J      DS   2
K      DC   2,0
ELMENTDC    2,0
ROW    DS   3
ELSTRTDS    5
I      DS   5
NX     DS   5
B      DSA  0
BFLAG  DSA  0
ASTADDDSA   0
SAVE   DS   84
WORKA  DSS  600
WORKB  DAS  100
       DAC  1,'
WORKC  DSS  100
RESET  DSC  50,0
       DC   50,0
       DC   34,0000000000000000000000000000000000'
DDA    DDA  ,1,0,1,WORKC
       DC   1,'
SMAIN  DD   ,DDA,,,A
DSKIN  DDA  ,0,0,0,WORKA
       DC   1,'
IN     DD   ,DSKIN,,,A
DSKOUTDDA   ,5,0,2,WORKB-1
       DC   1,'
OUT    DD   ,DSKOUT,,,A
LDISK  DAC  13,*LDISKRAP016'
RAP016DAC   29, SUBROUTINE RAP016 (BP,TR,P)'
DIM    DAC  35,DIMENSION BP(100),TR(100),P(50,10)'
RTURN  DAC  8, RETURN'
END    DAC  4,END'
CARD   DPRA,WORKB
*
*     TRANSFER OF ARGUMENTS
*
       DC   5,0
RAP004TFM   TF+6,NEL-4
       AM   TF+6,4,10
       AM   RAP004-1,5,10
       TF   CF+11,RAP004-1,11
       BNF  *+36,CF+11
CF     CF   CF+11
       TF   CF+11,CF+11,11
TF     TF   NEL,CF+11
       AM   TF+6,1,10
       BNR  RAP004+12,TF+6,11
       AM   RAP004-1,1,10
*
*     START OF SUBROUTINE
*
       TDM  SPEC,0
       CM   MCODE,0,68
       BE   *+32
```

Figure B-55. RAP004 Subprogram, Program Listing (2 of 12)

```
       TFM  DSKIN+5,10100
       B7   *+32
SKIP   TFM  DSKIN+5,10200
SPEC   DS   ,SKIP-1
       TDM  SPEC,1
       TDM  DSKIN,1
       TFM  DSKIN+8,1,9
       CF   RESET
       CF   RESET+50
       CF   RESET+100
       CF   LDISK-1
       CF   RAP016-1
       CF   DIM-1
       CF   RTURN-1
       CF   END-1
       TFM  DSKOUT+5,0
       TRNMWORKB+11,RESET
       TRNMWORKB-1,LDISK-1
       PUT  OUT
       TRNMWORKB-1,RESET
       TRNMWORKB+9,RAP016-1
       PUT  OUT
       BD   *+48,SPEC
       BNC4*+36
       PUT  CARD
       AM   DSKOUT+5,2,10
       TRNMWORKB+11,DIM-1
       PUT  OUT
       BD   *+48,SPEC
       BNC4*+36
       PUT  CARD
       AM   DSKOUT+5,2,10
*
*     GENERATE EQUATIONS FOR BASIC SUBMODE PROBABILITIES
*
       TRNMWORKB+11,RESET
       TFM  B,WORKB+12
       TFM  BFLAG,WORKB+11
       TFM  SMA,1
       TDM  NO,0
       TDM  REPEAT,0
       TFM  I,1,9
ALPHA  C    I,NROWS,11
       BH   ON
       GET  IN
       TFM  CONT,0,10
       TF   WORKB+10,CONT
       CF   WORKB+9
       TFM  J,1,10
SBACK  TF   ROW,I
       BTM  BPROB,*+12
       BD   FILL,NO
       BTM  EQUAL,*+12
       BTM  CKOUT,*+12
       BD   *+20,REPEAT
       B7   *+40
       TDM  NO,1
       TDM  REPEAT,0
       B7   SBACK
       TDM  TERM1,1
```

Figure B-55. RAP004 Subprogram, Program Listing (3 of 12)

```
        TFM DECODE+11,WORKA
GAMMA   C   J,NEL,11
        BNH DECODE
        AM  I,1,10
        AM  DSKIN+5,1,10
        BD  *+24,NEW
        BTM SEND,*+12
        TDM NEW,0
        B7  ALPHA
DECODEBNF BETA,WORKA
        AM  J,1,10
        \I  DECODE+11,1,10
        B7  GAMMA
BETA    BD  BETA+56,TERM1
        TF  ASTADD,BFLAG
        BTM AST,*+12
        BTM CKOUT,*+12
        B7  *+20
        TDM TERM1,0
        TF  ELMENT,J
        TFM K,0,10
        TD  K,DECODE+11,11
        AM  K,1,10
        BTM PROB,*+12
        B   DECODE+12
*
*     PLACE SAVED AREA FROM PREVIOUS LINE AT BEGINNING OF NEW LINE
*
FILL    TDM NO,0
        BD  FILBAK-12,OK
        A   B,NX
        SM  B,1,10
        TF  B,SAVE,6
        CF  BFLAG,,6
        AM  B,1,10
        TF  BFLAG,B
        AM  B,1,10
        B7  FILBAK
        TDM OK,0
FILBAKB7  CKOUT-1,,6
*
*     GENERATE EQUATIONS FOR SUBMODE PROBABILITY MINUS OVERLAP
*
ON      TFM ROW,0,9
        TFM COMROW+6,WORKA+2
        A   COMROW+5,NEL,11
        TDM SMA,0
        TDM DSKIN,3
        TFM DSKIN+5,19800
        TF  DSKIN+8,SECT,11
        TFM DDA+5,10100
        TFM I,1
LOOP    S   DSKIN+5,SECT,11
        TFM L1A+11,WORKC
        TF  L1+23,NST
        TFM ELSTRT,WORKA
        TFM B,WORKB+12
        TFM BFLAG,WORKB+11
        TFM CONT,0,10
        TF  WORKB+10,CONT
```

Figure B-55. RAP004 Subprogram, Program Listing (4 of 12)

```
         CF   WORKB+9
         TDM  NO,0
         TDM  TERM1,1
         TFM  ELMENT,1,10
COMPI    C    I,NARRAY,11
         BH   LAST
         GET  IN
         TFM  J,1,10
COMROWC       WORKA,ROW
         BE   EPS-12
         GET  SMAIN
         AM   DDA+5,1,10
         AM   ROW,1,10
         C    ROW,COMROW+6,11
         BE   NEXT
         BD   *+20,PATCH
         B7   PATAID
         TDM  ZRTN+1,9
         B7   NEXT
PATAIDTDM     PATCH,0
         TDM  ZRTN+1,1
         BTM  TROW,*+12
         BTM  EQUAL,*+12
         BTM  CKOUT,*+12
         BTM  ZERO,*+12
         BTM  CKOUT,*+12
         BTM  PERIOD,*+12
         BTM  CKOUT,*+12
         BTM  SEND,*+12
         TFM  CONT,0,10
         TF   WORKB+10,CONT
         CF   WORKB+9
         B7   COMROW+24
NEXT     CM   I,1,10
         BE   FIRST
         BD   *+20,PATCH
         B7   EPS-12
         SM   ROW,1,10
         BTM  TROW,*+12
         BTM  EQUAL,*+12
         BTM  CKOUT,*+12
         BTM  TROW,*+12
         BTM  AST,*+12
         BTM  CKOUT,*+12
         BTM  BPROB,*+12
         BTM  SEND,*+12
         TFM  CONT,0,10
         TF   WORKB+10,CONT
         CF   WORKB+9
         AM   ROW,1,10
ZRTN     NOP  PATAID
FIRST    TF   ROW,COMROW+6,11
         TDM  ROW1,1
         B7   *+20
         TDM  ROW1,0
EPS      BTM  X,*+12
         TDM  PATCH,1
         TDM  ZRTN+1,1
         BTM  CKOUT,*+12
         BD   FILL,NO
```

Figure B-55. RAP004 Subprogram, Program Listing (5 of 12)

```
        BTM  EQUAL,*+12
        BTM  CKOUT,*+12
        BD   *+20,REPEAT
        B7   *+40
        TDM  NO,1
        TDM  REPEAT,0
        B7   EPS
COMPJ   C    J,NEL,11
        BNH  L1A
        AM   I,1,10
        BD   DELTA-12,NEW
        CM   B,WORKB+16
        BNE  DELTA-24
        SF   WORKB+11
        CM   WORKB+14,6733,8
        CF   WORKB+11
        BNE  DELTA-24
        BTM  ONE,*+12
        BTM  CKOUT,*+12
        BTM  PERIOD,*+12
        BTM  CKOUT,*+12
        BTM  SEND,*+24
        TDM  NEW,0
DELTA   BTM  TROW,*+12
        BD   *+44,NO
        BTM  EQUAL,*+12
        BTM  CKOUT,*+12
        B7   *+52
        TDM  NO,0
        BTM  PLUS,*+12
        BTM  CKOUT,*+12
        B7   *+52
        BD   *+32,ROW1
        TDM  NO,1
        B7   DELTA
        TDM  ROW1,0
        BTM  X,*+12
        BTM  CKOUT,*+12
        TDM  WORKB+10,0
        TDM  WORKB+9,0
        BTM  SEND,*+12
        B7   LOOP
L1A     BNF  RETURN,WORKC
*
*     DETERMINE FORM OF TERM
*
L1      BTM  COUNT,*+12
        TF   *+35,NST
        AM   *+23,1,10
        CM   NONES,NST+1
        BNE  LOOK
RETURNAM     J,1,10
        AM   L1+23,4,10
        AM   ELMENT,1,10
        AM   ELSTRT,10
        AM   L1A+11,1,10
        B7   COMPJ
LOOK    A    NONES,NONES
        C    NONES,L1+47
        TDM  EL1,1
```

Figure B-55. RAP004 Subprogram, Program Listing (6 of 12)

```
        BH   NEG
        B7   POS
*
*     SUM OF TERMS
*
POS     TDM  INBRAC,1
        BD   *+48,TERM1
        TF   ASTADD,BFLAG
        BTM  AST,*+12
        BTM  CKOUT,*+12
        TDM  TERM1,0
        CM   NONES,2,10
        BE   *+36
        BTM  LPAR,*+12
        BTM  CKOUT,*+12
        TFM  K,1,10
        TF   *+23,ELSTRT
L3      BD   INSERT,ELSTRT
        C    K,L1+47
        BE   GO
        AM   K,1,10
        AM   L3+11,1,10
        B7   L3
INSERTBD     *+36,EL1
        BTM  PLUS,*+12
        BTM  CKOUT,*+12
        TDM  EL1,0
        BTM  PROB,*+12
        B7   L3+12
GO      CM   NONES,2,10
        BE   *+36
        BTM  RPAR,*+12
        BTM  CKOUT,*+12
        TDM  INBRAC,0
        B    RETURN
*
*     ONE MINUS THE SUM OF TERMS
*
NEG     TDM  INBRAC,1
        BD   *+48,TERM1
        TF   ASTADD,BFLAG
        BTM  AST,*+12
        BTM  CKOUT,*+12
        TDM  TERM1,0
        BTM  LPAR,*+12
        BTM  CKOUT,*+12
        BTM  ONE,*+12
        BTM  CKOUT,*+12
        BTM  PERIOD,*+12
        BTM  CKOUT,*+12
        TFM  K,1,10
        TF   *+23,ELSTRT
L5      BD   *+20,ELSTRT
        B7   L6
        C    K,L1+47
        BE   SHOW
        AM   K,1,10
        AM   L5+11,1,10
        B7   L5
L6      BTM  MINUS,*+12
```

Figure B-55. RAP004 Subprogram, Program Listing (7 of 12)

```
         BTM  CKOUT,*+12
         BTM  PROB,*+12
         B7   L5+20
SHOW     BTM  RPAR,*+12
         BTM  CKOUT,*+12
         TDM  INBRAC,0
         B    RETURN
*
*    INSERT P(ELMENT,K) INTO WORK AREA B
*
PROB     BTM  P,*+12
         BTM  CKOUT,*+12
         BTM  LPAR,*+12
         BTM  CKOUT,*+12
         BTM  ELNUM,*+12
         BTM  CKOUT,*+12
         BTM  COMMA,*+12
         BTM  CKOUT,*+12
         BTM  MODNUM,*+12
         BTM  CKOUT,*+12
         BTM  RPAR,*+12
         BTM  CKOUT,*+12
         B    PROB-1,,6
*
*    INSERT TR(ROW) INTO WORK AREA B
*
TROW     BTM  T,*+12
         BTM  CKOUT,*+12
         BTM  R,*+12
         BTM  CKOUT,*+12
         BTM  LPAR,*+12
         BTM  CKOUT,*+12
         BTM  RNUM,*+12
         BTM  CKOUT,*+12
         BTM  RPAR,*+12
         BTM  CKOUT,*+12
         B    TROW-1,,6
*
*    INSERT BP(ROW) INTO WORK AREA B
*
BPROB    BTM  BEE,*+12
         BTM  CKOUT,*+12
         BTM  P,*+12
         BTM  CKOUT,*+12
         BTM  LPAR,*+12
         BTM  CKOUT,*+12
         BTM  RNUM,*+12
         BTM  CKOUT,*+12
         BTM  RPAR,*+12
         BTM  CKOUT,*+12
         B    BPROB-1,,6
*
*    INSERT ELEMENT MODE NUMBER IN WORK AREA B
*
MODNUMCM      K,1,10
         BNE  *+24
         BT   ONE,MODNUM-1
         CM   K,2,10
         BNE  *+24
         BT   TWO,MODNUM-1
```

Figure B-55. RAP004 Subprogram, Program Listing (8 of 12)

```
         CM   K,3,10
         BNE  *+24
         BT   THREE,MODNUM-1
         CM   K,4,10
         BNE  *+24
         BT   FOUR,MODNUM-1
         CM   K,5,10
         BNE  *+24
         BT   FIVE,MODNUM-1
         CM   K,6,10
         BNE  *+24
         BT   SIX,MODNUM-1
         CM   K,7,10
         BNE  *+24
         BT   SEVEN,MODNUM-1
         CM   K,8,10
         BNE  *+24
         BT   EIGHT,MODNUM-1
         CM   K,9,10
         BNE  *+24
         BT   NINE,MODNUM-1
         BTM  ONE,*+12
         BTM  CKOUT,*+12
         BT   ZERO,MODNUM-1
*
*     COUNT THE AVAILABLE ELEMENT MODES
*
         DS   6
COUNT    TFM  NONES,0,10
         TF   THERE+11,ELSTRT
         TF   THERE+55,ELSTRT
         AM   THERE+55,9,10
THERE    BD   *+20,ELSTRT
         B7   *+20
         AM   NONES,1,10
         AM   THERE+11,1,10
         CM   THERE+11,ELSTRT
         BE   COUNT-1,,6
         B    THERE
*
*     INSERTION OF CHARACTERS IN WORK AREA B
*
ONE      TDM  BFLAG,7,6,       ONE
         TDM  B,1,6
         B    ONE-1,,6
TWO      TDM  BFLAG,7,6,       TWO
         TDM  B,2,6
         B    TWO-1,,6
THREE    TDM  BFLAG,7,6,       THREE
         TDM  B,3,6
         B    THREE-1,,6
FOUR     TDM  BFLAG,7,6,       FOUR
         TDM  B,4,6
         B    FOUR-1,,6
FIVE     TDM  BFLAG,7,6,       FIVE
         TDM  B,5,6
         B    FIVE-1,,6
SIX      TDM  BFLAG,7,6,       SIX
         TDM  B,6,6
         B    SIX-1,,6
```

Figure B-55. RAP004 Subprogram, Program Listing (9 of 12)

```
SEVEN  TDM BFLAG,7,6,       SEVEN
       TDM B,7,6
       B   SEVEN-1,,6
EIGHT  TDM BFLAG,7,6,       EIGHT
       TDM B,8,6
       B   EIGHT-1,8,6
NINE   TDM BFLAG,7,6,       NINE
       TDM B,9,6
       B   NINE-1,,6
ZERO   TDM BFLAG,7,6,       ZERO
       TDM B,0,6
       B   ZERO-1,,6
LPAR   TDM BFLAG,2,6,       LEFT PARENTHESIS
       TDM B,4,6
       B   LPAR-1,,6
RPAR   TDM BFLAG,0,6,       RIGHT PARENTHESIS
       TDM B,4,6
       B   RPAR-1,,6
AST    TDM BFLAG,1,6,       ASTERISK
       TDM B,4,6
       B   AST-1,,6
P      TDM BFLAG,5,6,       P
       TDM B,7,6
       B   P-1,,6
COMMA  TDM BFLAG,2,6,       COMMA
       TDM B,3,6
       B   COMMA-1,,6
PLUS   TDM BFLAG,1,6,       PLUS SIGN
       TDM B,0,6
       B   PLUS-1,,6
BEE    TDM BFLAG,4,6,       B
       TDM B,2,6
       B   BEE-1,,6
MINUS  TDM BFLAG,2,6,       MINUS SIGN
       TDM B,0,6
       B   MINUS-1,,6
PERIODTDM BFLAG,0,6,        PERIOD
       TDM B,3,6
       B   PERIOD-1,,6
EQUAL  TDM BFLAG,3,6,       EQUAL SIGN
       TDM B,3,6
       B   EQUAL-1,,6
T      TDM BFLAG,6,6,       T
       TDM B,3,6
       B   T-1,,6
X      TDM BFLAG,6,6,       X
       TDM B,7,6
       B   X-1,,6
R      TDM BFLAG,5,6,       R
       TDM B,9,6
       B   R-1,,6
*
*    INSERT ELEMENT NUMBER IN WORK AREA B
*
ELNUM  BD  *+20,ELMENT-1
       B7  *+56
       TDM BFLAG,7,6
       TD  B,ELMENT-1,6
       CF  B,,6
       BTM CKOUT,*+12
```

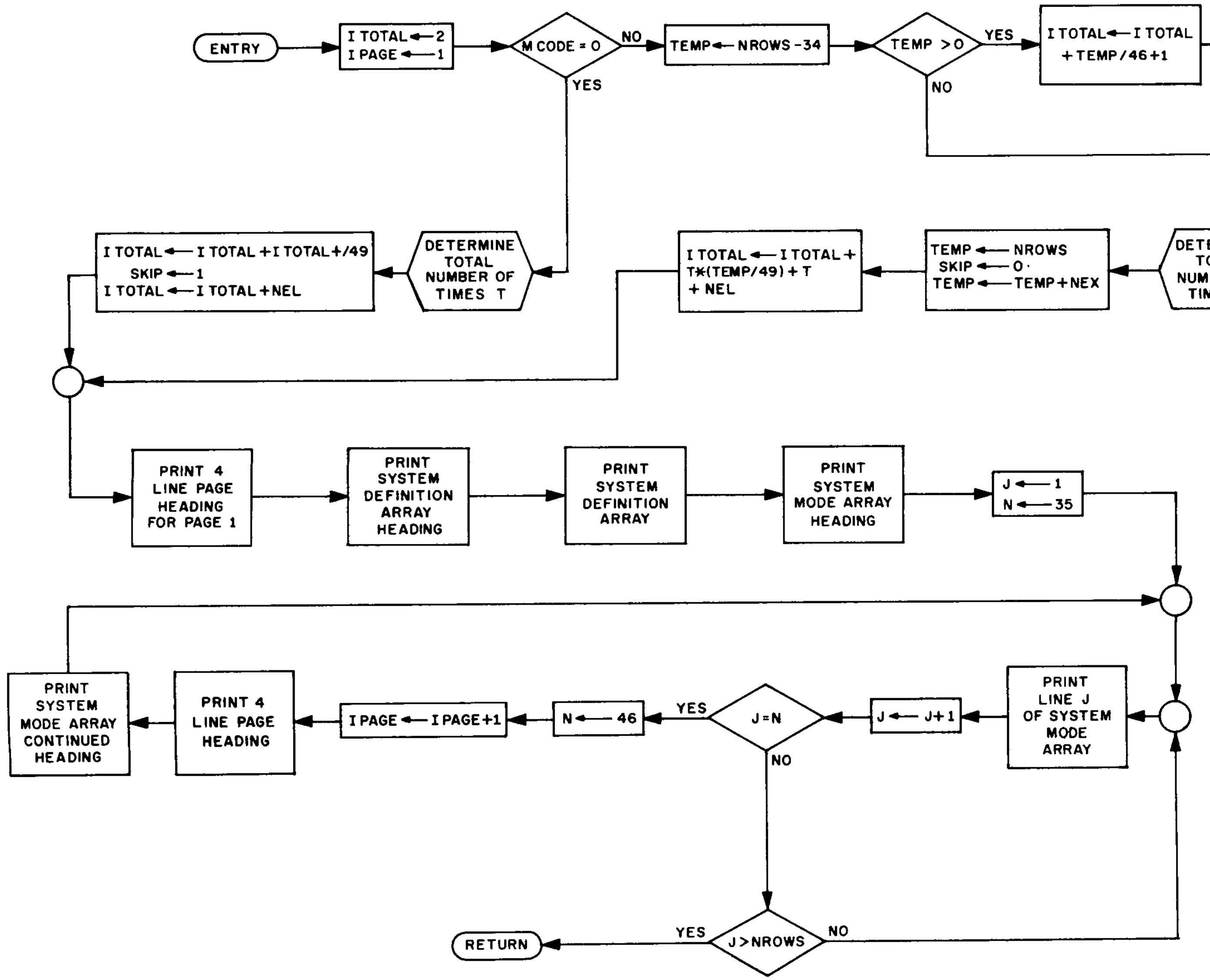
Figure B-55. RAP004 Subprogram, Program Listing (10 of 12)

```
       TDM BFLAG,7,6
       TD  B,ELMENT,6
       CF  B,,6
       B   ELNUM-1,,6
*
*      CHECK FOR EQUATION OUTPUT
*
CKOUT  CM  B,WORKB+142
       TDM NEW,0
       BE  *+48
       AM  B,2,10
       AM  BFLAG,2,10
       B   CKOUT-1,,6
       CM  CONT,4,10
       BE  SECOND
       BTM SEND,*+12
       B7  CKOUT-1,,6
SECONDTDM  REPEAT,1
       TF  *+66,BFLAG
       TF  *+66,B
       SM  *+42,2,10
       SM  *+42,2,10
       SF  BFLAG-2
       CM  B,64,10
       CF  *-18,,6
       BE  READY
       B7  NREADY
READY  BD  *+24,SMA
       BD  NREADY,INBRAC
       BTM SEND,*+12
       TFM CONT,0,10
       TF  WORKB+10,CONT
       CF  WORKB+9
       TF  FILBAK+6,CKOUT-1
       TDM OK,1
       BD  SBACK,SMA
       B7  EPS
NREADYBTM  SAVER,*+12
       TF  FILBAK+6,CKOUT-1
       TFM CONT,0,10
       TF  WORKB+10,CONT
       CF  WORKB+9
       TDM OK,0
       BD  SBACK,SMA
       B   EPS
*
*      OUTPUTS CARD IMAGE TO DISK STORAGE
*
SEND   PUT OUT
       BD  *+48,SPEC
       BNC4*+36
       PUT CARD
       TRNMWORKB+11,RESET
       TFM B,WORKB+12
       TFM BFLAG,WORKB+11
       AM  DSKOUT+5,2,10
       AM  CONT,1,10
       TF  WORKB+10,CONT
       TDM WORKB+9,7
       TDM NEW,1
       B   SEND-1,,6
*
*      END OF SUBROUTINE
*
```

Figure B-55. **RAP004 Subprogram, Program Listing (11 of 12)**

```
LAST   BTM TROW,*+12
       BTM EQUAL,*+12
       BTM CKOUT,*+12
       BTM TROW,*+12
       BTM AST,*+12
       BTM CKOUT,*+12
       BTM BPROB,*+12
       BTM SEND,*+12
LAST1  C   ROW,NROWS,11
       BE  FINISH
       AM  ROW,1,10
       BTM TROW,*+12
       BTM EQUAL,*+12
       BTM CKOUT,*+12
       BTM ZERO,*+12
       BTM CKOUT,*+12
       BTM PERIOD,*+12
       BTM CKOUT,*+12
       BTM SEND,*+12
       TFM CONT,0,10
       TF  WORKB+10,CONT
       CF  WORKB+9
       B7  LAST1
FINISHTRNMWORKB+9,RTURN-1
       PUT OUT
       BD  *+48,SPEC
       BNC4*+36
       PUT CARD
       AM  DSKOUT+5,2,10
       TRNMWORKB+11,RESET
       TRNMWORKB+11,END-1
       PUT OUT
       BD  *+48,SPEC
       BNC4*+36
       PUT CARD
       B   RAP004-1,,6
*
*
SAVER  SF  ASTADD,,6
       TF  SAVE,B,11
       TF  NX,B
       S   NX,ASTADD
       TF  *+18,ASTADD
       TDM ASTADD,0
       CM  *-6,WORKB+142
       BE  SAVER-1,,6
       AM  *-30,1,10
       B   *-48
*
*      INSERT ROW NUMBER IN WORK AREA B
*
RNUM   BD  *+20,ROW-2
       B7  *+56
       TDM BFLAG,7,6
       TD  B,ROW-2,6
       CF  B,,6
       BTM CKOUT,*+12
       BD  *+20,ROW-1
       B7  *+56
       TDM BFLAG,7,6
       TD  B,ROW-1,6
       CF  B,,6
       BTM CKOUT,*+12
       TDM BFLAG,7,6
       TD  B,ROW,6
       CF  B,,6
       B   RNUM-1,,6
BOTTOMDC   1,'
       DENDRAP004
```

Figure B-55. RAP004 Subprogram, Program Listing (12 of 12)



ARGUMENTS	
INPUT	OUTPUT
I PAGE	I PAGE
I TOTAL	
LD	
LM	
LN	
LS	
M CODE	
NEL	
NEX	
NIT	
NR	
NS	
NT	

Figure B-56. RAP005 Subprogram, Logic Flow Chart

```
*
*     RAP005 - - - PRINT OUT OF SDA AND SMA
*
*     SUBROUTINE LINKAGE
*
S       DS   ,*+101
        DC   6,987898,5-S
        DAC  6,RAP005,7-S
        DVLC22-S,5,LAST,2,08,2,04,5,RAP005-6,5,0,30,0
        DSC  17,0,0
        DORGS-100
LS      DSA  0
LN      DSA  0
LD      DSA  0
MCODE   DSA  0
LM      DSA  0
NST     DSA  0
NEL     DSA  0
NEX     DSA  0
NROWS   DSA  0
IPAGE   DSA  0
ITOTALDSA  0
NT      DSA  0
NIT     DSA  0
        DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*       I              INDEX
*       IPAGE          PAGE NUMBER
*       ITOTAL         TOTAL NUMBER OF PAGES
*       J              INDEX
*       K              INDEX
*       L              INDEX
*       LD(3)          DATE OF RUN
*       LM(23)         SYSTEM MODE NAME
*       LN(23)         SYSTEM NAME
*       LS(8)          SYSTEM CODE
*       MCODE          SYSTEM MODE NUMBER
*       NEL            NUMBER OF ELEMENTS IN SYSTEM
*       NEX            NUMBER OF SPACING LINES TO APPEAR ON INTERMEDIATE
*       NIT(25)        NUMBER OF INTERMEDIATE TIMES FOR EACH TRANSITION
*                      INTERVAL
*       NROWS          NUMBER OF ROWS IN SYSTEM MODE ARRAY
*       NST(50)        ENTRIES OF THE SYSTEM DEFINITION ARRAY
*       NT             NUMBER OF TRANSITION TIMES
*       NUMOR(50)      ARRAY CONTAINING THE NUMBER OF EACH ORDER
*       OUTPUT         OUTPUT AREA
*       SKIP           SWITCH
*       T              WORKING VARIABLE
*       T1             OUTPUT TITLE CONSTANT
*       T2             OUTPUT TITLE CONSTANT
*       T3             OUTPUT TITLE CONSTANT
*       T4             OUTPUT TITLE CONSTANT
*       T5             OUTPUT TITLE CONSTANT
*       T6             OUTPUT TITLE CONSTANT
*       T7             OUTPUT TITLE CONSTANT
*       T8             OUTPUT TITLE CONSTANT
*       T9             OUTPUT TITLE CONSTANT
*       T10            OUTPUT TITLE CONSTANT
```

Figure B-57. RAP005 Subprogram, Program Listing (1 of 8)

```
*       T11          OUTPUT  TITLE  CONSTANT
*       T12          OUTPUT  TITLE  CONSTANT
*       T13          OUTPUT  TITLE  CONSTANT
*       T14          OUTPUT  TITLE  CONSTANT
*       T15          OUTPUT  TITLE  CONSTANT
*       T16          OUTPUT  TITLE  CONSTANT
*       TEMP         WORKING VARIABLE
*       WORK         WORKING ARRAY
*       ZERO         OUTPUT  INITIALIZATION CONSTANT
*
TEMP    DC   4,0
I       DS   2
J       DS   3
K       DS   3
L       DS   2
SKIP    DS   1
T       DS   5
D       DDA  ,1,10000,1,NUMOR-1
        DC   1,'
DORDERDD     ,D,,,A
D1      DDA  ,1,10100,1,WORK
        DC   1,'
DSKIN   DD   ,D1,,,A
        DS   1
WORK    DSS  100
NUMOR   DAS  50
T1      DAC  8,RAPID-M1
T2      DAC  38,* PROBABILISTIC SYSTEM MODE ANALYSIS *
T3      DAC  4,PAGE
T4      DAC  2,OF
T5      DAC  4,DATE
T6      DAC  38,--*--*--*--*--*--*--*--*--*--*--*--
T7      DAC  6,SYSTEM
T8      DAC  16,DEFINITION ARRAY
T9      DAC  50,---------------------ELEMENTS---------------------
T10     DAC  50,12345678901234567890123456789012345678901234567890
T11     DAC  15,SPECIFIED MODES
T12     DAC  10,MODE ARRAY
T13     DAC  18,SUBMODE ( O/ N) **
T14     DAC  9,(   /   ) *
T15     DAC  22,MODE ARRAY (CONTINUED)
T16     DAC  2,**
OUTPUTDSS  160
        DAC  1,'
ZERO    DSC  50,0
        DC   50,0
        DC   50,0
        DC   10,0
        DAC  1,'
CARD    DPRA,OUTPUT+1
*
*     TRANSFER OF ARGUMENTS
*
        DC   5,0
RAP005TFM  TF+6,LS-4
        AM   TF+6,4,10
        AM   RAP005-1,5,10
        TF   CF+11,RAP005-1,11
        BNF  *+36,CF+11
CF      CF   CF+11
```

Figure B-57. RAP005 Subprogram, Program Listing (2 of 8)

```
        TF   CF+11,CF+11,11
TF      TF   LS,CF+11
        AM   TF+6,1,10
        BNR  RAP005+12,TF+6,11
        AM   RAP005-1,2,10
*
*     START OF SUBROUTINE
*
        TFM  D1+5,10100
INDEXIBBASBANDA
        BBBSBANDB
        TDM  GOODBY+11,0,,    SET BAND 0 EXIT
        B7   BANDO
BANDA   TDM  GOODBY+11,1,,    SET BAND 1 EXIT
        B7   BANDO
BANDB   TDM  GOODBY+11,2,,    SET BAND 2 EXIT
BANDO   BS   *+12,0,,         SELECT BAND 0 NOW
        TFM  IPAGE,0,68
        TF   BACK-13,NT,11
        SM   BACK-13,1,10
        TFM  ITOTAL,2,68
        CM   MCODE,0,68
        BNE  NORMAL
        TDM  SKIP,1
        BTM  TIMES,*+12
        LD   97,T
        DM   95,49,10
        A    ITOTAL,95,6
        A    ITOTAL,NEL,611
        B7   BEGIN
NORMALTF     TEMP,NROWS,11
        SM   TEMP,34,10
        CM   TEMP,0,10
        BH   *+20
        B7   *+56
        AM   ITOTAL,1,610
        LD   97,TEMP
        DM   95,46,10
        A    ITOTAL,95,6
        BTM  TIMES,*+12
        TF   TEMP,NROWS,11
        A    TEMP,NEX,11
        LD   97,TEMP
        DM   95,49,10
        TF   *+23,95
        MM   T,0
        A    ITOTAL,99,6
        A    ITOTAL,NEL,611
        A    ITOTAL,T,6
        TDM  SKIP,0
BEGIN   GET  DORDER
        BTM  HEAD,*+12
*
*     PRINT SYSTEM DEFINITION ARRAY HEADING
*
        TR   OUTPUT,ZERO
        TF   OUTPUT+85,T7+10
        PUT  CARD
        TF   OUTPUT+95,T8+30
        PUT  CARD
```

Figure B-57. RAP005 Subprogram, Program Listing (3 of 8)

```
        BTM T90UT,*+12
*
*      PRINT SYSTEM DEFINITION ARRAY
*
        BTM T16OUT,*+12
        TR  OUTPUT,ZERO
        TFM OUTPUT+149,14,10
        TFM OUTPUT+43,14,10
        TF  OUTPUT+39,T11+28
        TF  ELMODE+11,NST
        TFM I,1,10
        TFM ELMODE+6,OUTPUT+47
        TFM ELMODE+18,OUTPUT+46
ELMODETD    OUTPUT+47,NST
        TDM OUTPUT+46,7
        AM  I,1,10
        AM  ELMODE+6,2,10
        AM  ELMODE+11,4,10
        AM  ELMODE+18,2,10
        C   I,NEL,11
        BNH ELMODE
        PUT CARD
        BTM T16OUT,*+12
        TR  OUTPUT,ZERO
        PUT CARD
        TF  OUTPUT+153,T6+74
        TF  OUTPUT+81,T6+74
        PUT CARD
        BD  GOODBY,SKIP
*
*      PRINT SYSTEM MODE ARRAY HEADING
*
        TR  OUTPUT,ZERO
        TF  OUTPUT+85,T7+10
        PUT CARD
        TF  OUTPUT+89,T12+18
        PUT CARD
        BTM T9OUT,*+12
        TR  OUTPUT,ZERO
        TF  OUTPUT+149,T16+2
        TF  OUTPUT+45,T16+2
        TF  OUTPUT+45,T13+34
        PUT CARD
*
*      PRINT SYSTEM MODE ARRAY
*
        TFM JLOOK+11,35,9
        TR  OUTPUT,ZERO
SMA     TFM J,1,9
        TFM I,1,10
        TFM *+35,NUMOR
        TFM K,1,10
LOOP1   C   K,NUMOR
        BNH LINE
        AM  I,1,10
        AM  *-25,2,10
        B7  SMA+36
LINE    TR  OUTPUT,ZERO
        GET DSKIN
        TFM *+59,WORK
```

Figure B-57. RAP005 Subprogram, Program Listing (4 of 8)

```
         TFM  ENTRY+6,OUTPUT+46
         TFM  ENTRY+18,OUTPUT+47
         TFM  L,1,10
         BNF  ENTRY,WORK
         AM   *-1,1,10
         AM   L,1,10
         AM   ENTRY+6,2,10
         AM   ENTRY+18,2,10
         TDM  ENTRY+6,0,6
         TDM  ENTRY+18,0,6
         C    L,NEL,11
         BNH  LINE+84
         B7   ENTRY+32
ENTRY    TDM  OUTPUT+46,7
         TD   OUTPUT+47,LINE+95,11
         B7   LINE+96
         TD   OUTPUT+17,J
         TDM  OUTPUT+16,7
         BD   *+20,J-1
         B7   *+32
         TD   OUTPUT+15,J-1
         TDM  OUTPUT+14,7
         BD   *+20,J-2
         B7   *+32
         TD   OUTPUT+13,J-2
         TDM  OUTPUT+12,7
         TF   OUTPUT+43,T14+16
         TD   OUTPUT+37,K
         TDM  OUTPUT+36,7
         BD   *+20,K-1
         B7   *+32
         TD   OUTPUT+35,K-1
         TDM  OUTPUT+34,7
         TD   OUTPUT+31,I
         TDM  OUTPUT+30,7
         BD   *+20,I-1
         B7   *+32
         TD   OUTPUT+29,I-1
         TDM  OUTPUT+28,7
         TFM  OUTPUT+149,14,10
         PUT  CARD
         AM   K,1,10
         AM   J,1,10
         AM   D1+5,1,10
JLOOK    CM   J,35,10
         BE   *+20
         B7   *+20
         BTM  EXTRA,*+12
         C    J,NROWS,11
         BNH  LOOP1
         BTM  T16OUT,*+12
         TR   OUTPUT,ZERO
         PUT  CARD
         TF   OUTPUT+153,T6+74
         TF   OUTPUT+81,T6+74
         PUT  CARD
GOODBYBS      RAP005-1,,6
         DS   6
*
*     PRINT SYSTEM MODE ARRAY CONTINUATION HEADING
```

Figure B-57. RAP005 Subprogram, Program Listing (5 of 8)

```
*
EXTRA BTM  HEAD,*+12
      AM   JLOOP+11,46,10
      TR   OUTPUT,ZERO
      TF   OUTPUT+85,T7+10
      PUT  CARD
      TF   OUTPUT+101,T15+42
      PUT  CARD
      BTM  T9OUT,*+12
      B    EXTRA-1,,6
*
*     CALCULATE TOTAL NUMBER OF TIMES
*
TIMES TFM  I,1,10
      TFM  TEMP,0,6
      TF   BACK+11,NIT
      CM   I,NT-1
      BH   TRANS
BACK  A    TEMP,NIT
      AM   BACK+11,4,10
      AM   I,1,10
      B7   BACK-24
TRANS A    TEMP,NT,11
      TF   T,TEMP
      B    TIMES-1,,6
*
*     OUTPUT T9 AND T10
*
T9OUT TR   OUTPUT,ZERO
      PUT  CARD
      TF   OUTPUT+145,T9+98
      PUT  CARD
      TR   OUTPUT,ZERO
      TFM  I,1,10
      TFM  *+30,OUTPUT+65
      TFM  *+30,OUTPUT+64
      TD   OUTPUT+65,I
      TDM  OUTPUT+64,7
      AM   I,1,10
      AM   *-30,20,10
      AM   *-30,20,10
      CM   I,5,10
      BNH  *-72
      PUT  CARD
      TF   OUTPUT+145,T10+98
      PUT  CARD
      B    T9OUT-1,,6
*
*     OUTPUT T16
*
T16OUTTR   OUTPUT,ZERO
      TF   OUTPUT+149,T16+2
      TF   OUTPUT+45,T16+2
      PUT  CARD
      B    T16OUT-1,,6
*
*     PRINT TOP OF PAGE
*
HEAD  TR   OUTPUT,ZERO
      SKIP,1
```

Figure B-57. RAP005 Subprogram, Program Listing (6 of 8)

```
          AM   IPAGE,1,610
          TF   OUTPUT+21,T1+14
          TF   OUTPUT+117,T2+74
          TF   OUTPUT+135,T3+6
          TF   TEMP,IPAGE,11
          BD   *+20,TEMP-2
          B7   *+32
          TDM  OUTPUT+136,7
          TD   OUTPUT+137,TEMP-2
          BD   *+20,TEMP-1
          B7   *+32
          TDM  OUTPUT+138,7
          TD   OUTPUT+139,TEMP-1
          BD   *+20,TEMP
          B7   *+32
          TDM  OUTPUT+140,7
          TD   OUTPUT+141,TEMP
          TF   OUTPUT+147,T4+2
          TF   TEMP,ITOTAL,11
          BD   *+20,TEMP-2
          B7   *+32
          TDM  OUTPUT+148,7
          TD   OUTPUT+149,TEMP-2
          BD   *+20,TEMP-1
          B7   *+32
          TDM  OUTPUT+150,7
          TD   OUTPUT+151,TEMP-1
          BD   *+20
          B7   *+32
          TDM  OUTPUT+152,7
          TD   OUTPUT+153,TEMP
          PUT  CARD
          TR   OUTPUT,ZERO
          TFM  I,1,10
          TF   CODE+11,LS
          TFM  CODE+6,OUTPUT+7
          SM   CODE+11,2,10
CODE      TF   OUTPUT+7,LS
          AM   CODE+6,2,10
          AM   CODE+11,4,10
          AM   I,1,10
          CM   I,8,10
          BNH  CODE
          TFM  I,1,10
          TF   NAME+11,LN
          TFM  NAME+6,OUTPUT+37
NAME      TF   OUTPUT+37,LN
          AM   NAME+6,4,10
          AM   NAME+11,4,10
          AM   I,1,10
          CM   I,23,10
          BNH  NAME
          TFM  I,1,10
          TF   DATE+11,LD
          TFM  DATE+6,OUTPUT+141
DATE      TF   OUTPUT+141,LD
          AM   DATE+6,6,10
          AM   DATE+11,4,10
          AM   I,1,10
          CM   I,3,10
```

Figure B-57. RAP005 Subprogram, Program Listing (7 of 8)

```
         BNH  DATE
         BD   MONTH,OUTPUT+140
         TDM  OUTPUT+140,7
         B7   *+44
MONTH    TD   OUTPUT+139,OUTPUT+140
         TDM  OUTPUT+138,7
         TDM  OUTPUT+140
         BD   DAY,OUTPUT+146
         TDM  OUTPUT+146,7
         B7   *+44
DAY      TD   OUTPUT+145,OUTPUT+146
         TDM  OUTPUT+144,7
         TDM  OUTPUT+146,7
         BD   YEAR,OUTPUT+152
         TDM  OUTPUT+152,7
         B7   *+44
YEAR     TD   OUTPUT+151,OUTPUT+152
         TDM  OUTPUT+152,7
         TDM  OUTPUT+150,7
         TF   OUTPUT+135,T5+6
         TFM  OUTPUT+143,21,10
         TFM  OUTPUT+149,21,10
         PUT  CARD
         TR   OUTPUT,ZERO
         TFM  I,1,10
         TF   MODE+11,LM
         TFM  MODE+6,OUTPUT+37
MODE     TF   OUTPUT+37,LM
         AM   MODE+6,4,10
         AM   MODE+11,4,10
         AM   I,1,10
         CM   I,23,10
         BNH  MODE
         PUT  CARD
         TR   OUTPUT,ZERO
         TF   OUTPUT+153,T6+74
         TF   OUTPUT+81,T6+74
         PUT  CARD
         B7   HEAD-1,,6
LAST     DC   2,'
         DENDRAP005
```

Figure B-57. RAP005 Subprogram, Program Listing (8 of 8)

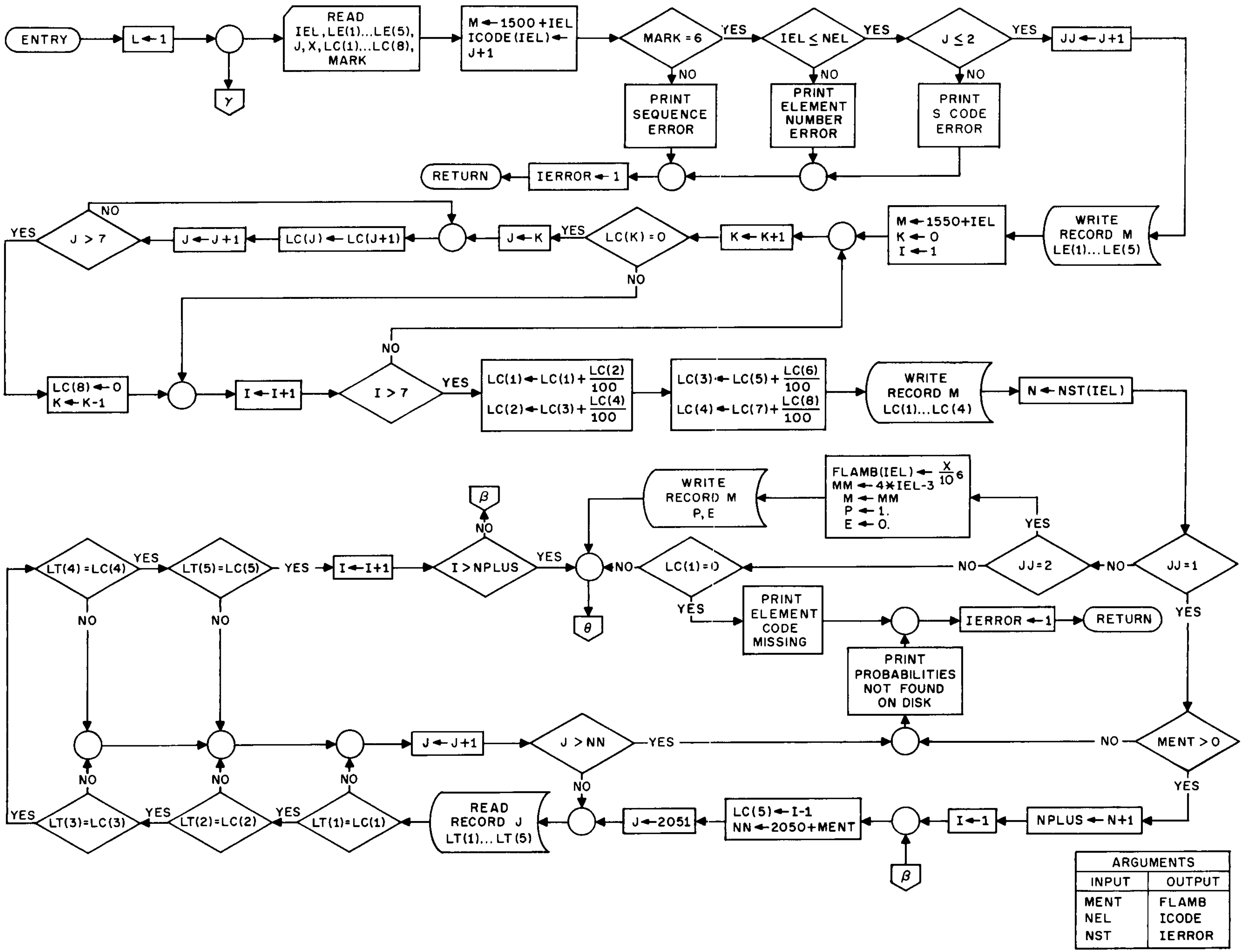


Figure B-58. RAP007 Subprogram, Logic Flow Chart (1 of 2)

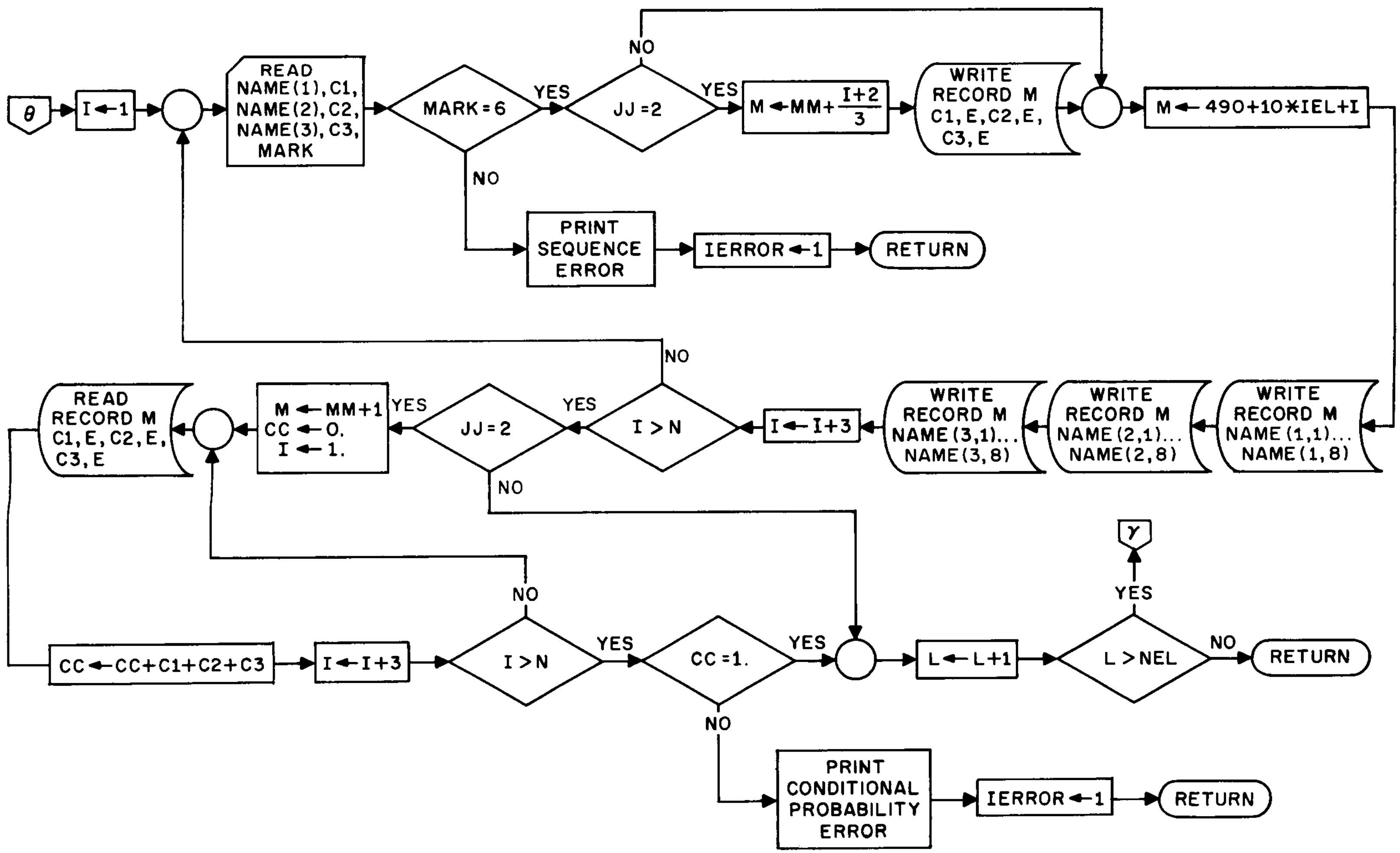


Figure B-58. RAP007 Subprogram, Logic Flow Chart (2 of 2)

```
C
C        RAP007 - - - INPUT FROM DATA TRANSMITTAL FORM III
C
C     C1          CONDITIONAL PROBABILITY
C     C2          CONDITIONAL PROBABILITY
C     C3          CONDITIONAL PROBABILITY
C     CC          SUM OF CONDITIONAL PROBABILITIES
C     E           EXPONENT
C     FLAMB(50)   FAILURE RATES PER HOUR
C     I           INDEX
C     ICODE(50)   ARRAY CONTAINING S CODE
C     IEL         ELEMENT NUMBER
C     IERROR      INDICATOR, 0 IF NO INPUT ERROR, 1 IF AN INPUT ERROR OCCURRED
C     J           INDEX
C     JFIX        INDEX
C     JJ          INDEX
C     K           INDEX
C     L           INDEX
C     LC(8)       ELEMENT CODE
C     LE(5)       ELEMENT NAME
C     LT(8)
C     M           INDEX
C     MARK        I.D. NO. IN COL. 80 OF INPUT CARDS
C     MENT        NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
C     MM          INDEX
C     N           INDEX
C     NAME(3,8)   ELEMENT MODE NAMES
C     NEL         NUMBER OF ELEMENTS IN SYSTEM
C     NN          INDEX
C     NPLUS       NUMBER OF ELEMENT MODES
C     NST(50)     ENTRIES OF THE SYSTEM DEFINITION ARRAY
C     P           PROBABILITY
C     X           FAILURE RATE PER MILLION HOURS
C
      SUBROUTINE RAP007 (NEL,NST,ICODE,FLAMB,MENT,IERROR)
      DIMENSION LE(5),LC(8),NAME(3,8),NST(50),FLAMB(50),ICODE(50),LT(8)
  100 FORMAT (1X,I2,1X,5A2,I2,1X,E10.2,2X,8A1,41X,I2)
  101 FORMAT (3X,3(2X,8A2,1X,E6.1),I2)
  102 FORMAT (17H DATA SET IGNORED)
  103 FORMAT (1H ,I2,21H CARD OUT OF SEQUENCE)
  104 FORMAT (1H ,35HELEMENT NUMBER ERROR IN TYPE 6 CARD)
  105 FORMAT (1H ,12HS CODE ERROR)
  106 FORMAT (1H ,18HPROBABILITIES FOR ,4A2,5H MODE,I3,12H NOT ON DISK)
  107 FORMAT (1H ,42HCONDITIONAL PROBABILITY ERROR FOR ELEMENT ,I2)
  108 FORMAT( 1H ,32HELEMENT CODE MISSING FOR ELEMENT,I3)
      DO 38 L=1,NEL
      READ 100,IEL,LE,J,X,LC,MARK
      M=1500+IEL
      ICODE(IEL)=J+1
      IF(MARK-6)31,5,31
    5 IF (NEL-IEL)32,6,6
    6 IF (J-2)7,7,33
    7 JJ=J+1
      RECORD (M) (LE(K),K=1,5)
      M=1550+IEL
      FIND (M)
C
C        COMPRESS ELEMENT CODE
C
      K=0
```

Figure B-59. RAP007 Subprogram, Program Listing (1 of 3)

```
      DO 4 I=1,7
      K=K+1
      IF (LC(K))4,2,4
    2 DO 3 J=K,7
    3 LC(J)=LC(J+1)
      LC(8)=0
      K=K-1
    4 CONTINUE
      LC(1)=LC(1)+LC(2)/100
      LC(2)=LC(3)+LC(4)/100
      LC(3)=LC(5)+LC(6)/100
      LC(4)=LC(7)+LC(8)/100
      RECORD (M) (LC(I),I=1,4)
      N=NST(IEL)
      GO TO(8,16,40),JJ
C
C          LOCATE ELEMENT PROBABILITIES ON DISK
C
   40 IF(LC(1))17,41,17
    8 IF(MENT)34,34,9
    9 NPLUS=N+1
      DO 15 I=1,NPLUS
      LC(5)=I-1
      NN=2050+MENT
      DO 14 J=2051,NN
      JFIX=J
      FETCH (JFIX) (LT(K),K=1,5)
      IF(LT(1)-LC(1))14,10,14
   10 IF(LT(2)-LC(2))14,11,14
   11 IF(LT(3)-LC(3))14,12,14
   12 IF(LT(4)-LC(4))14,13,14
   13 IF(LT(5)-LC(5))14,15,14
   14 CONTINUE
      GO TO 34
   15 CONTINUE
      GO TO 17
   16 FLAMB(IEL)=X/1000000.
      MM=4*IEL-3
      M=MM
      P=1.
      E=0.
      RECORD (M) P,E
   17 DO 1 I=1,N,3
C
C          READ ELEMENT MODE NAMES AND CONDITIONAL PROBABILITIES
C
      READ 101,(NAME(1,K),K=1,8),C1,(NAME(2,K),K=1,8),C2,(NAME(3,K),K=1,
     18),C3,MARK
      IF (MARK-6)31,39,31
   39 GO TO (19,18,19),JJ
   18 M=MM+(I+2)/3
      RECORD (M) C1,E,C2,E,C3,E
   19 M=490+10*IEL+I
      RECORD (M) (NAME(1,K),K=1,8)
      RECORD (M) (NAME(2,K),K=1,8)
    1 RECORD (M) (NAME(3,K),K=1,8)
C
C          CHECK SUM OF CONDITIONAL PROBABILITIES
C
      IF (JJ-2)38,36,38
```

Figure B-59. RAP007 Subprogram, Program Listing (2 of 3)

```
   36 M=MM+1
      CC=0.
      DO 37 I=1,N,3
      FETCH (M) C1,E,C2,E,C3,E
   37 CC=CC+C1+C2+C3
      IF(CC-1.)35,38,35
   38 CONTINUE
      RETURN
C
C           CARD SEQUENCE ERROR
C
   31 PRINT 103,MARK
   30 IERROR=1
      PRINT 102
      RETURN
C
C           ERROR IN ELEMENT NUMBER
C
   32 PRINT 104
      GO TO 30
C
C           ERROR IN S CODE
C
   33 PRINT 105
      GO TO 30
C
C           PROBABILITIES MISSING
C
   34 PRINT 106,(LC(J),J=1,5)
      GO TO 30
C
C           ERROR IN CONDITIONAL PROBABILITIES
C
   35 PRINT 107,IEL
      GO TO 30
   41 PRINT 108,IEL
      GO TO 30
      END
```

Figure B-59. RAP007 Subprogram, Program Listing (3 of 3)

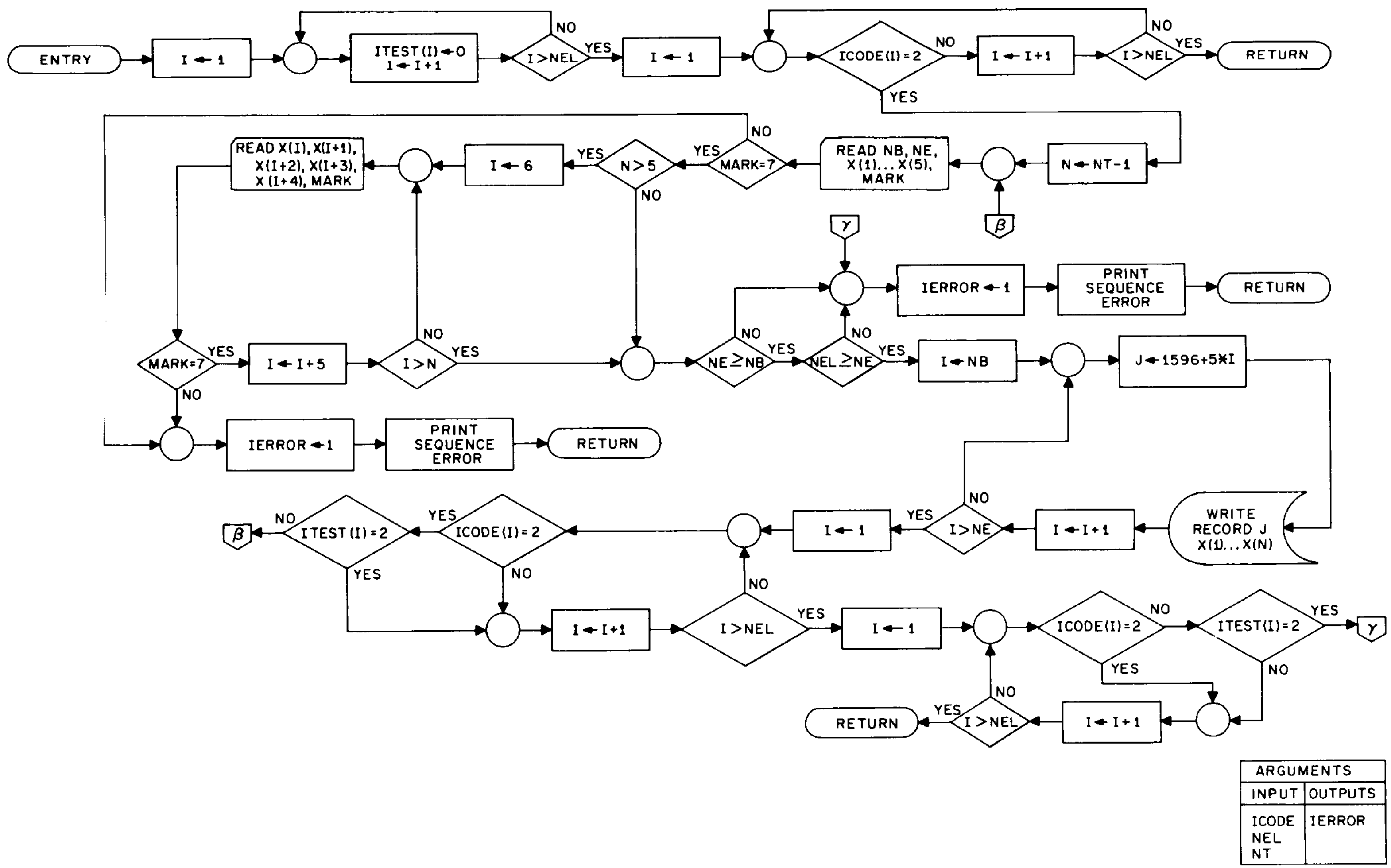
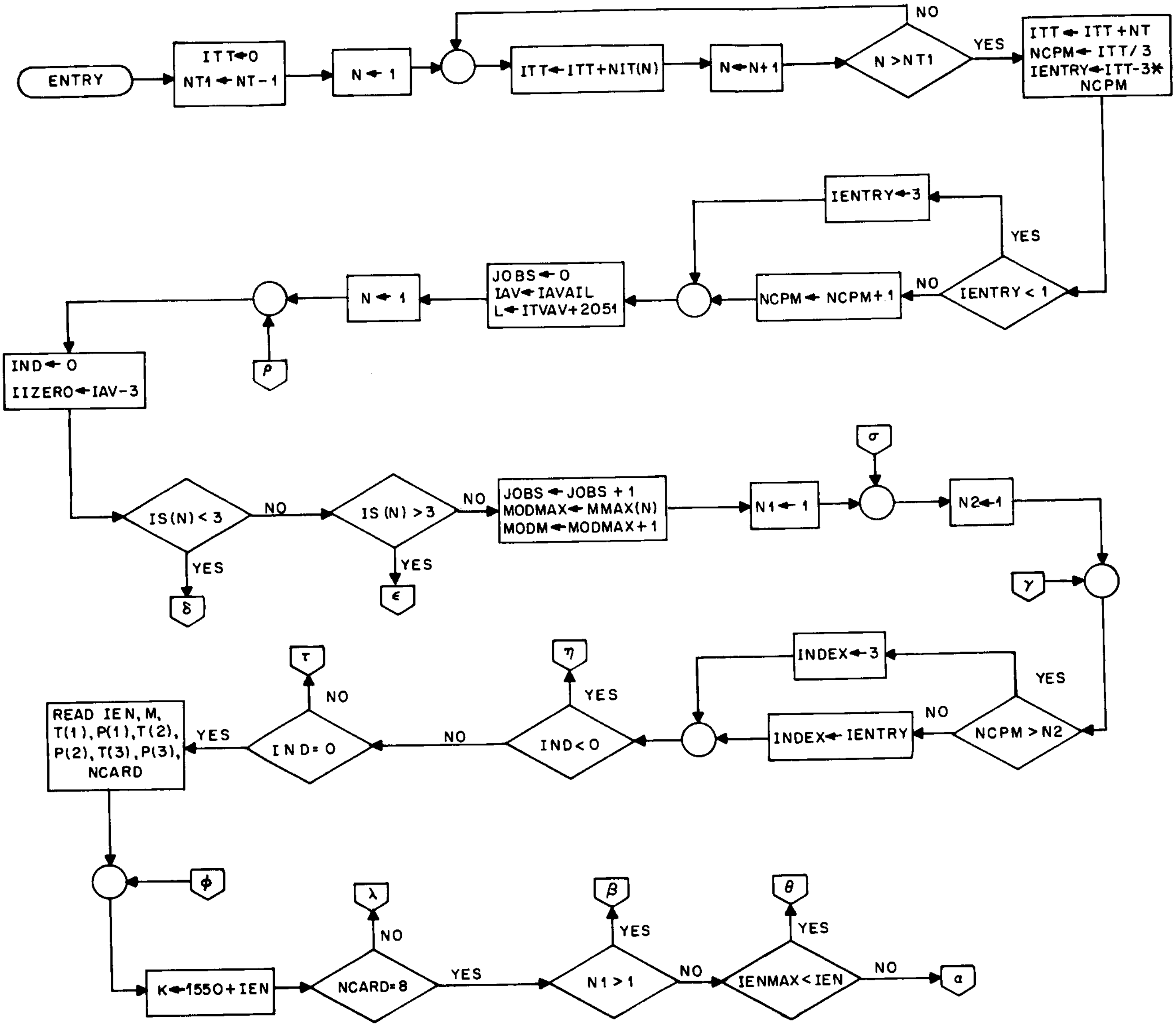


Figure B-60. RAP008 Subprogram, Logic Flow Chart

```
C
C          RAP008 - - - INPUT FROM DATA TRANSMITTAL FORM III A
C
C
C     I              INDEX
C     ICODE(50)      ARRAY CONTAINING S CODE
C     ITEST(50)      ARRAY FOR KEEPING TRACK OF ELEMENTS PROCESSED
C     IERROR         INDICATOR, 0 IF NO INPUT ERROR, 1 IF AN INPUT ERROR OCCURRED
C     J              INDEX
C     MARK           I.D. NO. IN COL. 80 OF INPUT CARDS
C     N              NUMBER OF TRANSITION INTERVALS
C     NB             BEGINNING ELEMENT NUMBER
C     NE             ENDING ELEMENT NIMBER
C     NEL            NUMBER OF ELEMENTS IN SYSTEM
C     NT             NUMBER OF TRANSITION TIMES
C     X(25)          K FACTORS
C
      SUBROUTINE RAP008 (NEL,NT,ICODE,IERROR)
      DIMENSION X(25),ICODE(50),ITEST(50)
  100 FORMAT(1X,I2,2X,I2,5(2X,E10.2,2X),I3)
  101 FORMAT(7X,5(2X,E10.2,2X),I3)
  102 FORMAT (1H ,I2,21H CARD OUT OF SEQUENCE/1H ,25HDATA SET IGNORED -
  1021RAP008)
  103 FORMAT (1H ,35HELEMENT NUMBER ERROR IN TYPE 7 CARD/1H ,25HDATA SET
  1031 IGNORED - RAP008)
      DO 20 I=1,NEL
   20 ITEST(I)=0
      DO 21 I=1,NEL
      IF (ICODE(I)-2)21,22,21
   21 CONTINUE
      RETURN
   22 N=NT-1
    1 READ 100,NB,NE,(X(I),I=1,5),MARK
      IF(MARK-7)14,2,14
    2 IF(N-5)5,5,3
    3 DO 4 I=6,N,5
      READ 101,X(I),X(I+1),X(I+2),X(I+3),X(I+4),MARK
      IF(MARK-7)14,4,14
    4 CONTINUE
    5 IF(NE-NB)15,6,6
    6 IF(NEL-NE)15,7,7
    7 DO 8 I=NB,NE
      J=1596+5*I
      ITEST(I)=2
    8 RECORD (J) (X(K),K=1,N)
      DO 11 I=1,NEL
      IF (ICODE(I)-2)11,10,11
   10 IF (ITEST(I)-2)1,11,1
   11 CONTINUE
      DO 13 I=1,NEL
      IF (ICODE(I)-2)12,13,12
   12 IF (ITEST(I)-2)13,15,13
   13 CONTINUE
      RETURN
C
C          CARD SEQUENCE ERROR
C
   14 IERROR=1
      PRINT 102,MARK
      RETURN
C
C          ERROR IN ELEMENT NUMBER
C
   15 IERROR=1
      PRINT 103
      RETURN
      END
```

Figure B-61. RAP008 Subprogram, Program Listing



ARGUMENTS	
INPUT	OUTPUT
IAVAIL	IAVAIL
ICODE	IERROR
MENT	MENT
NEL	
NIT	
NST	
NT	

Figure B-62. RAP009 Subprogram, Logic Flow Chart (1 of 4)

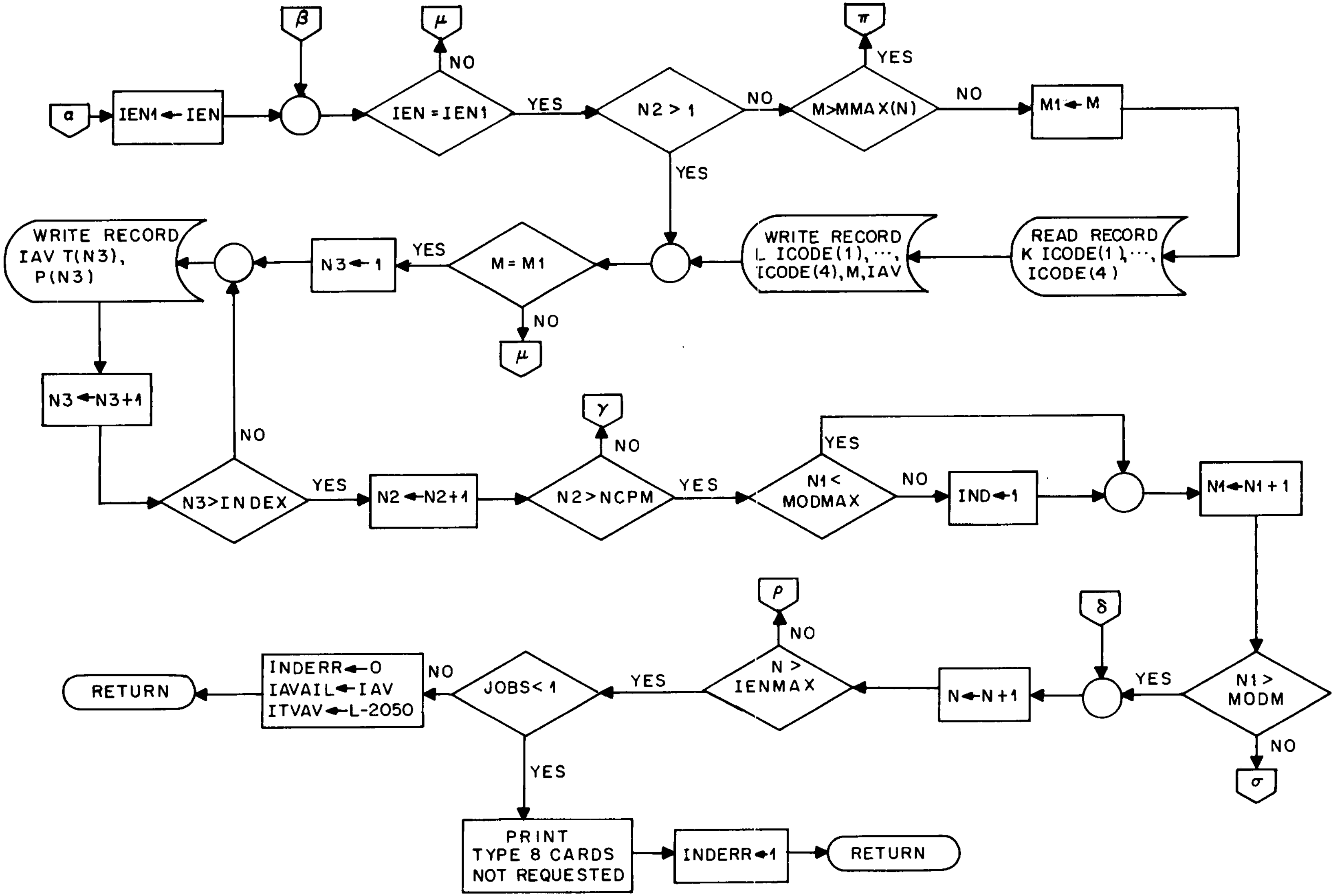


Figure B-62. RAP009 Subprogram, Logic Flow Chart (2 of 4)

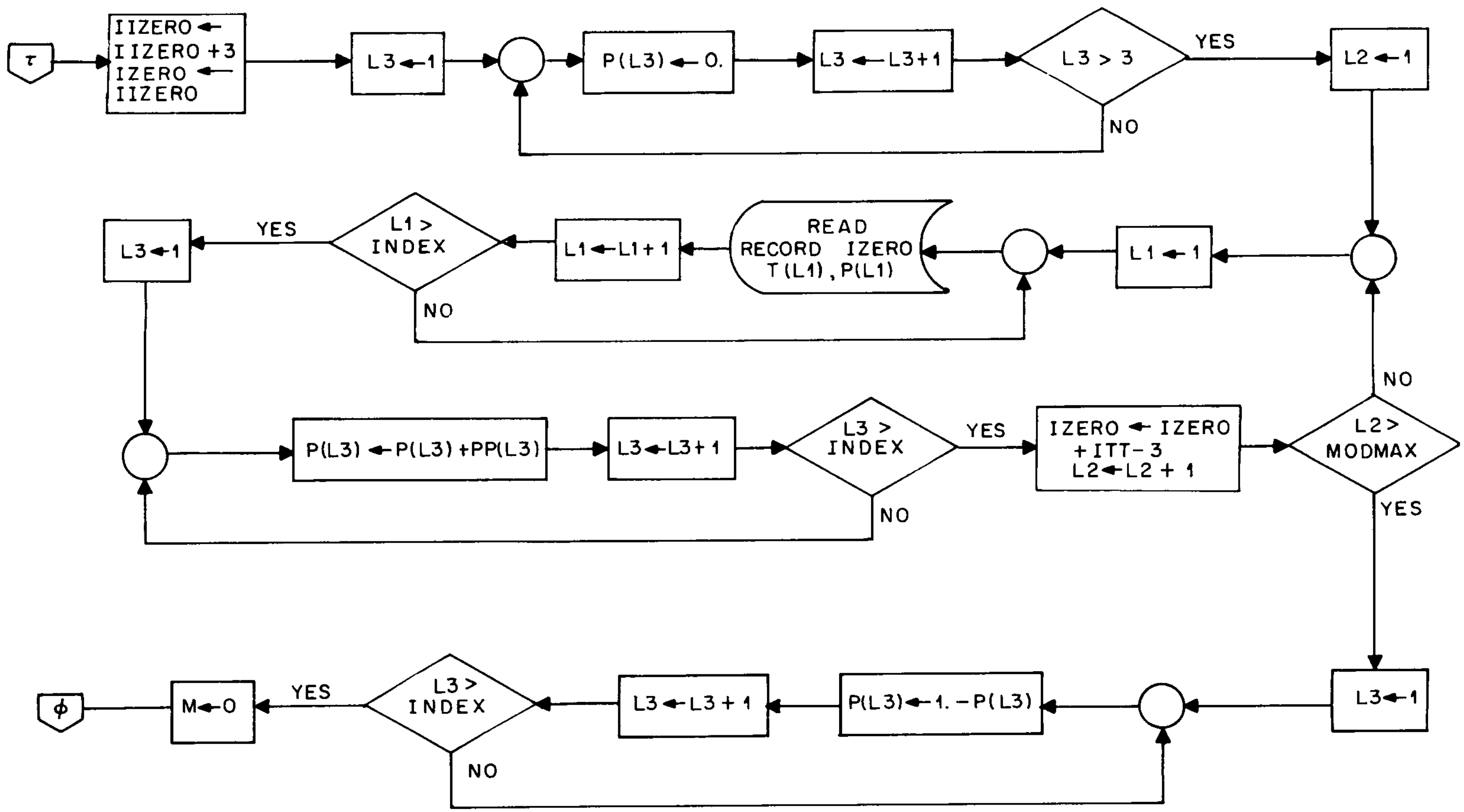


Figure B-62. RAP009 Subprogram, Logic Flow Chart (3 of 4)

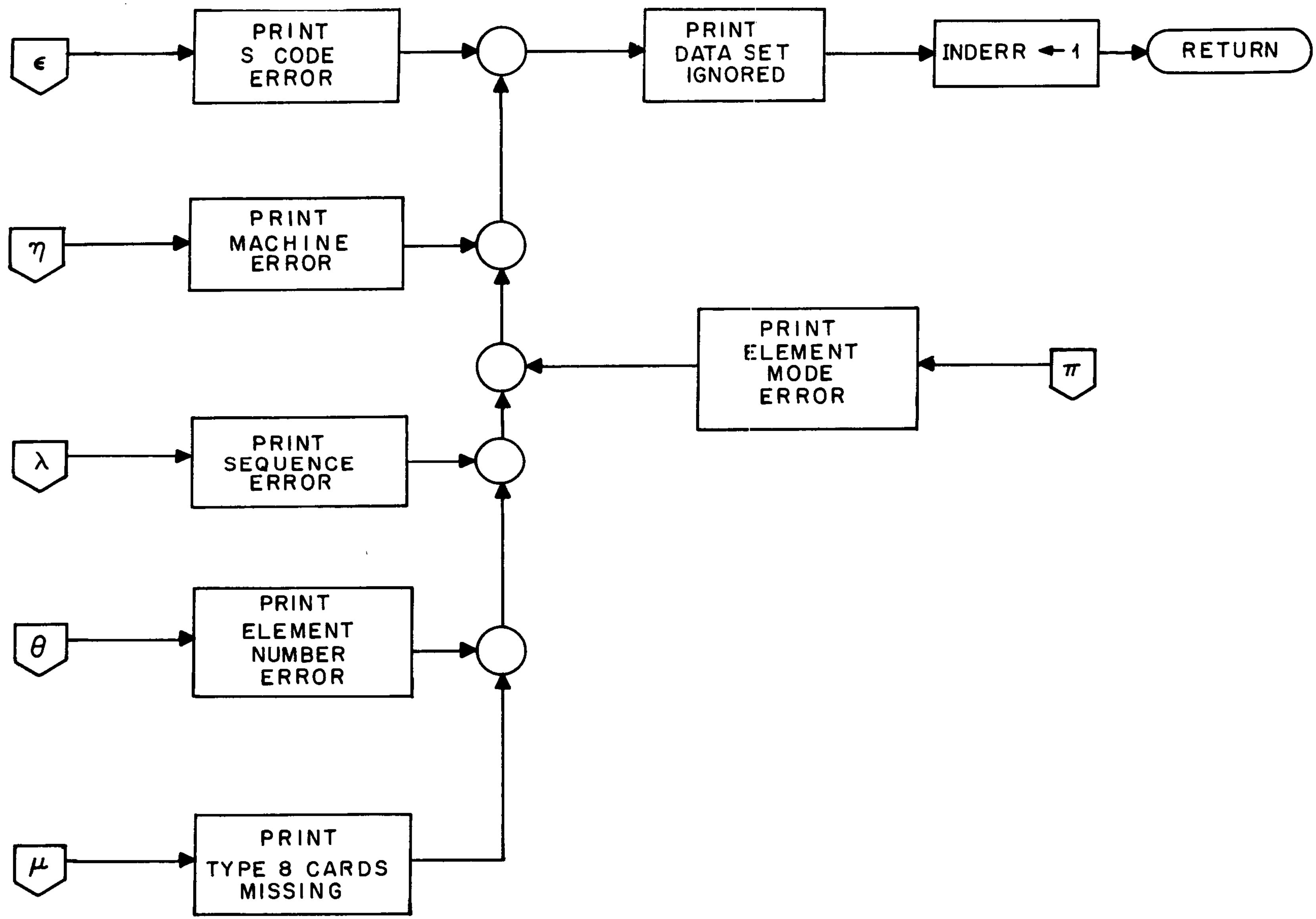


Figure B-62. RAP009 Subprogram, Logic Flow Chart (4 of 4)

```
C
C       RAP009 - - - INPUT FROM DATA TRANSMITTAL FORM III B
C
C
C      IAV           INITIAL AVAILABLE RECORD NUMBER
C      IAVAIL        CURRENT AVAILABLE RECORD NUMBER
C      ICODE(4)      ELEMENT CODE
C      IEN           ELEMENT NUMBER
C      IEN1          SAVED ELEMENT NUMBER
C      IENMAX        NUMBER OF ELEMENTS
C      IENTRY        NUMBER OF ENTRIES PER MODE
C      IIZERO        INDEX
C      IND           MODE ZERO SWITCH
C      INDERR        INDICATOR, 0 IF NO INPUT ERROR, 1 IF AN INPUT ERROR OCCURRED
C      INDEX         NUMBER OF ENTRIES ON NEXT CARD
C      IS(50)        ARRAY CONTAINING S CODE
C      ITT           TOTAL NUMBER OF TIMES
C      ITVAV         NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
C      IZERO         INDEX
C      JOBS          NUMBER OF PASSES
C      K             INDEX
C      L             INDEX
C      L1            INDEX
C      L2            INDEX
C      L3            INDEX
C      M             MODE NUMBER
C      M1            SAVED MODE NUMBER
C      MMAX(50)      ENTRIES OF THE SYSTEM DEFINITION ARRAY
C      MODM          NUMBER OF MODES FOR CURRENT ELEMENT
C      MODMAX        MAXIMUM MODE NUMBER FOR CURRENT ELEMENT
C      N             INDEX
C      N1            INDEX
C      N2            INDEX
C      N3            INDEX
C      NCARD         I.D. NO. IN COL. 80 OF INPUT CARDS
C      NCPM          NUMBER OF CARDS PER MODE
C      NIT(25)       NUMBER OF INTERMEDIATE TIMES FOR EACH TRANSITION INTERVAL
C      NT            NUMBER OF TRANSITION TIME
C      NT1           NUMBER OF TRANSITION INTERVALS
C      P(3)          PROBABILITY SUMMATION
C      PP(3)         INPUT PROBABILITIES
C      T(3)          TIME
C
      SUBROUTINE RAP009(IS,IENMAX,MMAX,NT,NIT,IAVAIL,ITVAV,INDERR)
      DIMENSION T(3),P(3),PP(3),ICODE(4),MMAX(50),NIT(25),IS(50)
  100 FORMAT(1H ,13HMACHINE ERROR/26H DATA SET IGNORED - RAP009)
  101 FORMAT(2I3,3(F12.0,F12.0),I2)
  102 FORMAT(1H ,I2,21H CARD OUT OF SEQUENCE)
  103 FORMAT(1H ,35HELEMENT NUMBER ERROR IN TYPE 8 CARD)
  104 FORMAT(1H ,31HTYPE 8 CARD MISSING FOR ELEMENT,I3)
  105 FORMAT(1H ,22HMODE ERROR FOR ELEMENT,I3)
  106 FORMAT(1H ,31HTYPE 8 CARD MISSING FOR ELEMENT,I3,5H MODE,I3)
  107 FORMAT(26H DATA SET IGNORED - RAP009)
  109 FORMAT(1H ,12HS CODE ERROR)
  110 FORMAT(52H INPUT OF TYPE 8 CARDS NOT REQUESTED FOR ANY ELEMENT)
      ITT=0
      NT1=NT-1
      DO 1 N=1,NT1
    1 ITT=ITT+NIT(N)
      ITT=ITT+NT
```

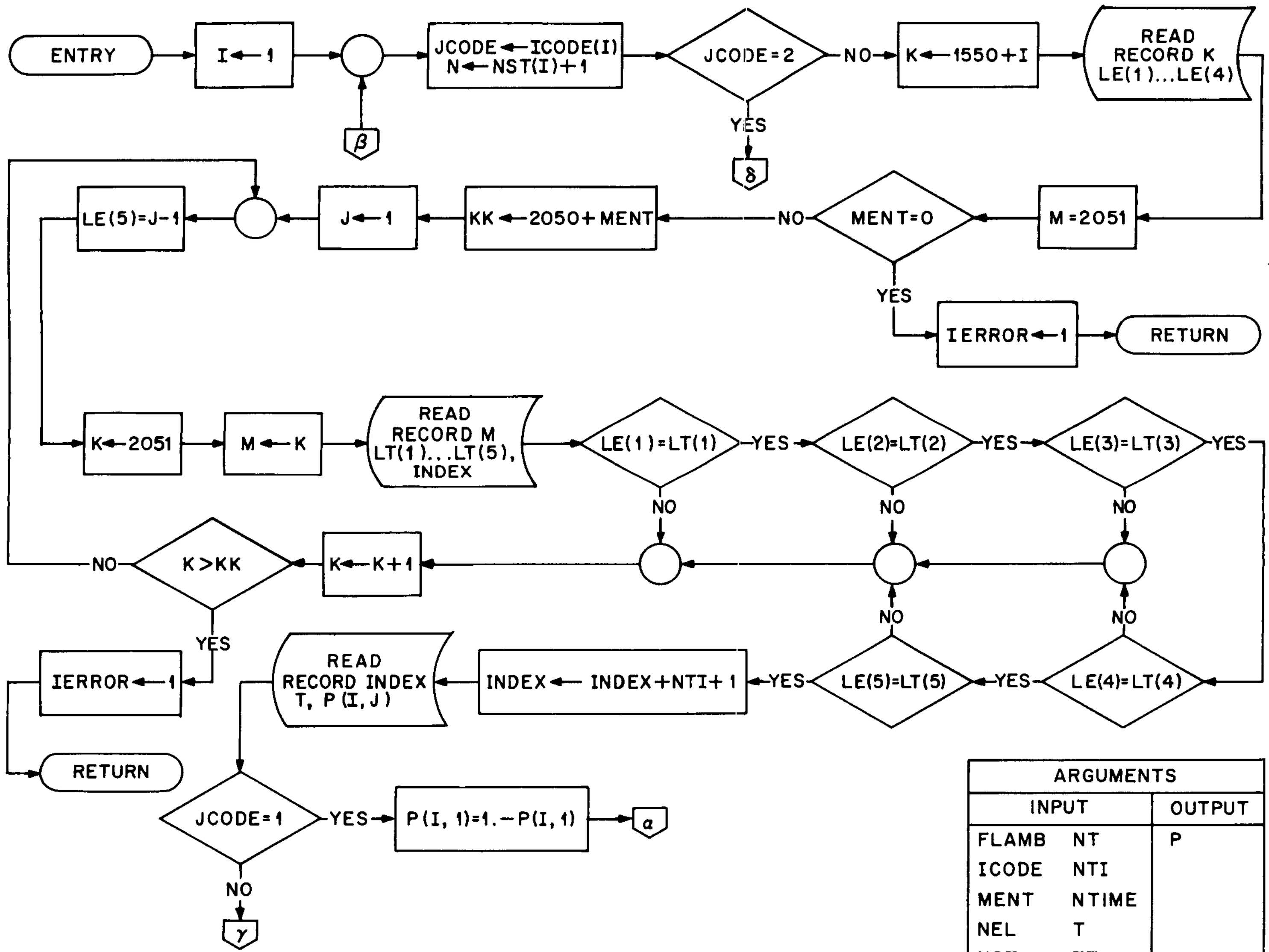
Figure B-63. RAP009 Subprogram, Program Listing (1 of 3)

```
      NCPM=ITT/3
      IENTRY=ITT-3*NCPM
      IF(IENTRY-1)99,2,2
   99 IENTRY=3
      GO TO 3
    2 NCPM=NCPM+1
    3 JOBS=0
      IAV=IAVAIL
      L=ITVAV+2051
    4 DO 50 N=1,IENMAX
      IND=0
      IIZERO=IAV-3
      IF(IS(N)-3)50,6,7
    7 PRINT 109,IS(N)
      GO TO 98
    6 JOBS=JOBS+1
      MODMAX=MMAX(N)
      MODM=MODMAX+1
    9 DO 51 N1=1,MODM
      DO 52 N2=1,NCPM
   25 IF(NCPM-N2)26,26,27
   26 INDEX=IENTRY
      GO TO 75
   27 INDEX=3
   75 IF(IND)70,71,72
   70 PRINT 100
      GO TO 98
   71 READ 101,IEN,M,(T(I),P(I),I=1,3),NCARD
   73 K=1550+IEN
      FIND(K)
      IF(NCARD-8)10,11,10
   10 PRINT 102,NCARD
      GO TO 98
   72 IIZERO=IIZERO+3
      IZERO=IIZERO
      DO 63 L3=1,3
   63 P(L3)=0.
      DO 62 L2=1,MODMAX
      DO 61 L1=1,INDEX
   61 FETCH (IZERO) T(L1),PP(L1)
      DO 64 L3=1,INDEX
   64 P(L3)=P(L3)+PP(L3)
   62 IZERO=IZERO+IIT-3
      DO 65 L3=1,INDEX
   65 P(L3)=1.-P(L3)
      M=0
      GO TO 73
   11 IF(N1-1)12,12,13
   12 IF(IENMAX-IEN)14,15,15
   14 PRINT 103
      GO TO 98
   15 IEN1=IEN
   13 IF(IEN-IEN1)16,17,16
   16 PRINT 104,IEN1
      GO TO 98
   17 IF(N2-1)18,18,19
   18 IF(MMAX(N)-M)20,21,21
   20 PRINT 105,IEN
      GO TO 98
   21 M1=M
```

Figure B-63. RAP009 Subprogram, Program Listing (2 of 3)

```
24 FETCH(K)  (ICODE(KK),KK=1,4)
   RECORD(L)  (ICODE(LL),LL=1,4),M,IAV
19 IF(M-M1)22,28,22
22 PRINT 106, IEN,M1
   GO TO 98
28 DO 53 N3=1,INDEX
53 RECORD(IAV)  T(N3),P(N3)
52 CONTINUE
   IF  (N1-MODMAX)51,74,74
74 IND=1
51 CONTINUE
50 CONTINUE
   IF(JOBS-1)29,30,30
29 PRINT 110
   INDERR=1
   RETURN
30 INDERR=0
   IAVAIL=IAV
   ITVAV=L-2050
   RETURN
98 PRINT 107
   INDERR=1
   RETURN
   END
```

Figure B-63. RAP009 Subprogram, Program Listing (3 of 3)



ARGUMENTS		
INPUT		OUTPUT
FLAMB	NT	P
ICODE	NTI	
MENT	NTIME	
NEL	T	
NST	TT	

Figure B-64. RAP010 Subprogram, Logic Flow Chart (1 of 2)

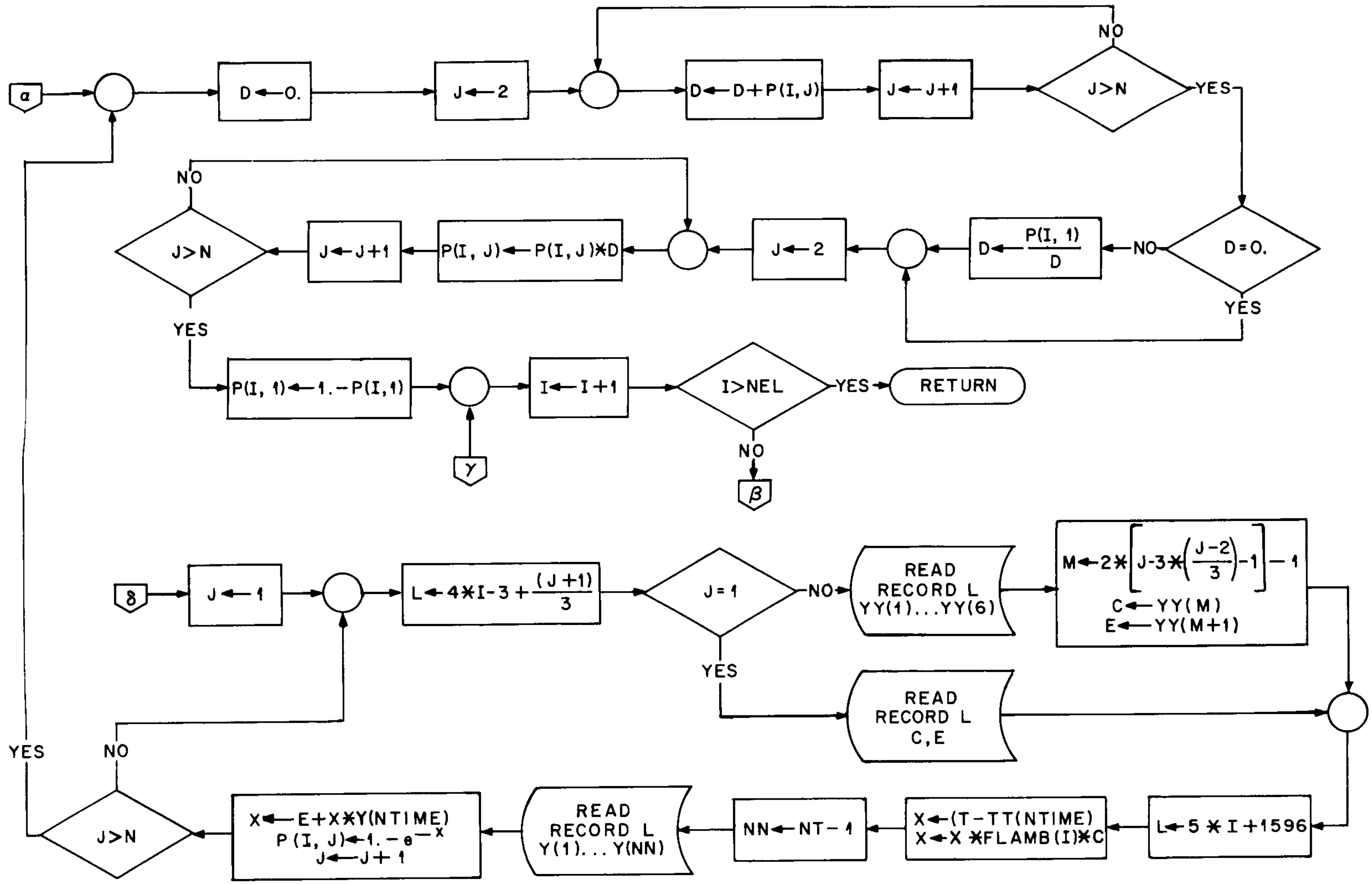


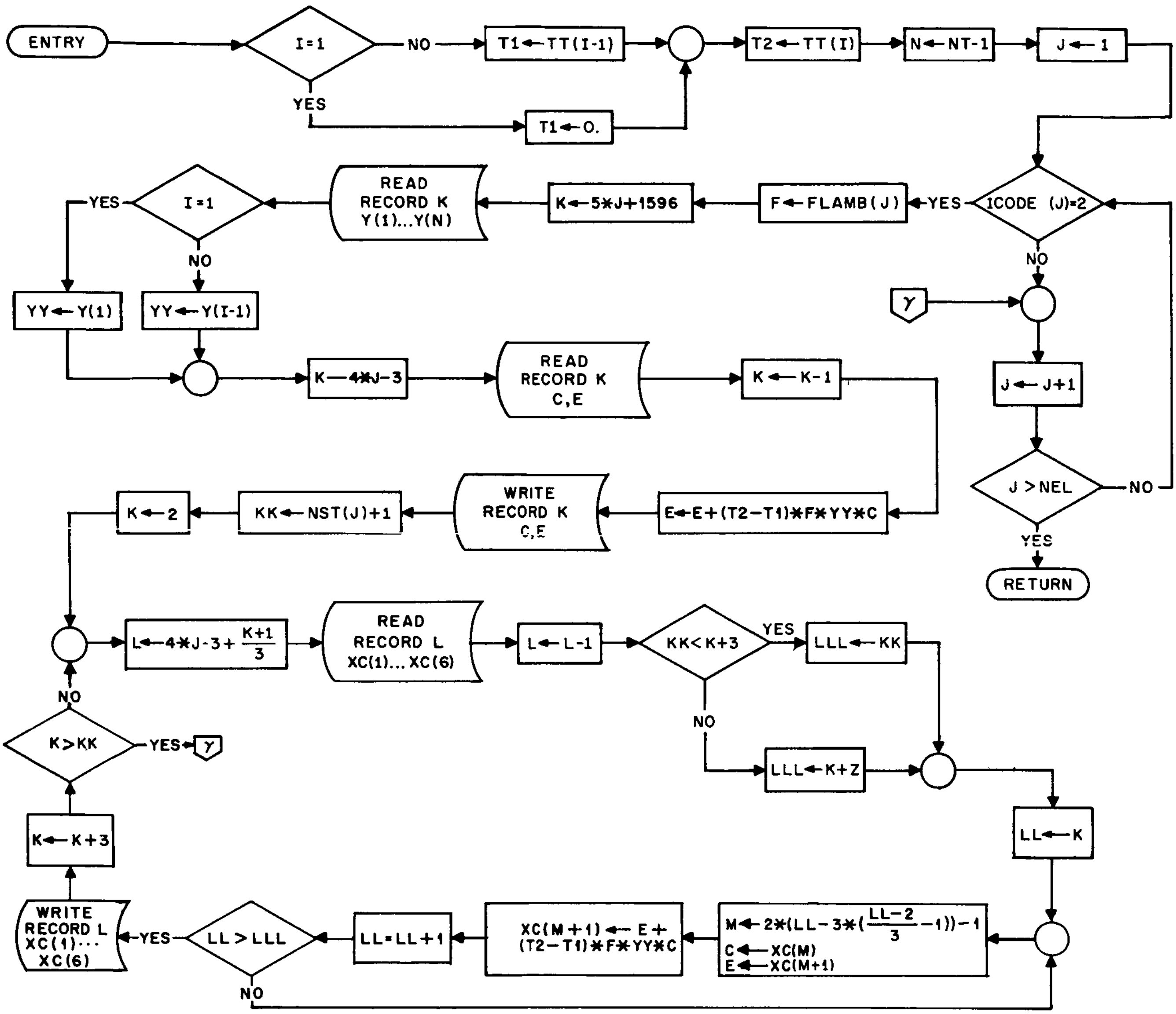
Figure B-64. RAP010 Subprogram, Logic Flow Chart (2 of 2)

```
C
C        RAPO10 - - - CALCULATE ELEMENT MODE PROBABILITIES
C
C
C    C           CONDITIONAL PROBABILITY
C    D           WORKING VARIABLE
C    E           VALUE OF EXPONENT UP TO LAST TRANSITION TIME
C    FLAMB(50)   ARRAY CONTAINING FAILURE RATES PER HOUR
C    I           INDEX
C    ICODE(50)   ARRAY CONTAINING S CODE
C    IERROR      INDICATOR, 0 IF NO INPUT ERROR, 1 IF AN INPUT ERROR OCCURRED
C    INDEX       INDEX
C    J           INDEX
C    JCODE       S CODE FOR CURRENT ELEMENT
C    K           INDEX
C    KK          INDEX
C    L           INDEX
C    LE(5)       ELEMENT CODE AND MODE
C    LT(5)       TEMPORARY WORKING ARRAY
C    M           INDEX
C    MENT        NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
C    N           NUMBER OF MODES FOR CURRENT ELEMENT
C    NEL         NUMBER OF ELEMENTS IN SYSTEM
C    NN          NUMBER OF TRANSITION INTERVALS
C    NST(50)     ENTRIES OF THE SYSTEM DEFINITION ARRAY
C    NT          NUMBER OF TRANSITION TIMES
C    NTI         TIME INDEX FOR STORED PROBABILITIES
C    NTIME       INDEX TO LAST TRANSITION TIME
C    P(50,10)    ELEMENT MODE PROBABILITY
C    T           CURRENT TIME
C    TT(25)      TRANSITION TIMES
C    X           VALUE OF EXPONENT AT CURRENT TIME
C    Y(25)       ARRAY CONTAINING K FACTORS
C    YY(6)       TEMPORARY WORKING ARRAY
C
      SUBROUTINE RAPO10 (NEL,NST,T,TT,FLAMB,NT,ICODE,NTI,MENT,NTIME,P)
      DIMENSION LE(5),LT(5),Y(25),NST(50),TT(25),FLAMB(50),ICODE(50),P(5
     10,10)
      DIMENSION YY(6)
      DO 20 I=1,NEL
      JCODE=ICODE(I)
      N=NST(I)+1
      GO TO(1,11,1),JCODE
    1 K=1550+I
      FETCH(K)(LE(L),L=1,4)
      M=2051
      FIND(M)
      IF(MENT)2,8,2
    2 KK=2050+MENT
      DO 10 J=1,N
      LE(5)=J-1
      DO 7 K=2051,KK
      M=K
      FETCH(M) (LT(L),L=1,5),INDEX
      IF(LE(1)-LT(1))7,3,7
    3 IF(LF(2)-LT(2))7,4,7
```

Figure B-65. RAP010 Subprogram, Program Listing (1 of 2)

```
 4 IF(LF(3)-LT(3))7,5,7
 5 IF(LF(4)-LT(4))7,6,7
 6 IF(LF(5)-LT(5))7,9,7
 7 CONTINUE
 8 IERROR=1
   RETURN
 9 INDEX=INDEX+NTI-1
10 FETCH (INDEX)T,P(I,J)
   GO TO(24,20,20),JCODE
24 P(I,1)=1.-P(I,1)
   GO TO 15
11 DO 14 J=1,N
   L=4*I-3+(J+1)/3
   IF(J-1)22,21,22
21 FETCH (L)C,E
   GO TO 23
22 FETCH(L) (YY(M),M=1,6)
   M=2*(J-3*((J-2)/3)-1)-1
   C=YY(M)
   E=YY(M+1)
23 L=5*I+1596
   FIND(L)
   X=(T-TT(NTIME))*FLAMB(I)*C
   NN=NT-1
   FETCH(L) (Y(M),M=1,NN)
   X=E+X*Y(NTIME)
   IF(X-1.4)12,12,13
12 P(I,J)=X*(1.-X/2.*(1.-X/3.*(1.-X/4.*(1.-X/5.*(1.-X/6.*(1.-X/7.*(1.
  1-X/8.*(1.-X/9.*(1.-X/10.*(1.-X/11.*(1.-X/12.*(1.-X/13.*(1.-X/14.*(
  21.-15.*(1.-X/16.)))))))))))))))
   GO TO 14
13 P(I,J)=1.-EXPF(-X)
14 CONTINUE
15 D=0.
   DO 16 J=2,N
16 D=D+P(I,J)
   IF(D)17,18,17
17 D=P(I,1)/D
18 DO 19 J=2,N
19 P(I,J)=P(I,J)*D
   P(I,1)=1.-P(I,1)
20 CONTINUE
   RETURN
   END
```

Figure B-65. RAP010 Subprogram, Program Listing (2 of 2)

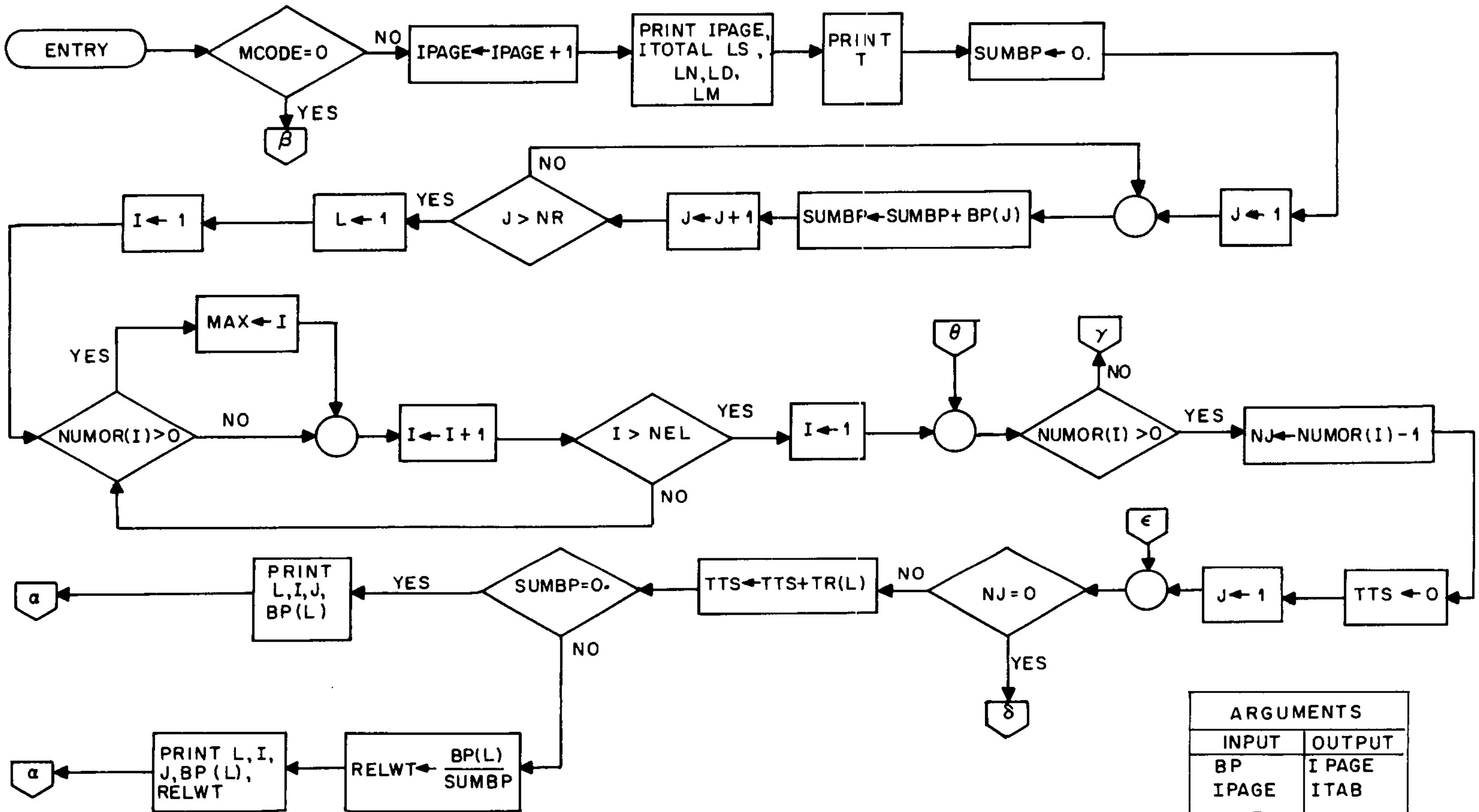


ARGUMENTS	
INPUT	OUTPUT
FLAMB	
I	
ICODE	
NEL	
NST	
NT	
TT	

Figure B-66. RAP011 Subprogram, Logic Flow Chart

```
C
C        RAP011 - - - UPDATE EXPONENT FOR E TO THE MINUS LAMBDA T
C
C
C     C            CONDITIONAL PROBABILITY
C     E            VALUE OF EXPONENT UP TO LAST TRANSITION TIME
C     F            FAILURE RATE PER HOUR FOR CURRENT ELEMENT
C     FLAMB(50)    ARRAY CONTAINING FAILURE RATES PER HOUR
C     I            INDEX
C     ICODE(50)    ARRAY CONTAINING S CODE
C     J            INDEX
C     K            INDEX
C     KK           INDEX
C     L            INDEX
C     LL           INDEX
C     LLL          INDEX
C     M            INDEX
C     N            NUMBER OF TRANSITION INTERVALS
C     NEL          NUMBER OF ELEMENTS IN SYSTEM
C     NST(50)      ENTRIES OF THE SYSTEM DEFINITION ARRAY
C     NT           NUMBER OF TRANSITION TIMES
C     T1           PREVIOUS TRANSITION TIME
C     T2           CURRENT TRANSITION TIME
C     TT(25)       TRANSITION TIMES
C     XC(6)        WORKING ARRAY
C     Y(25)        ARRAY CONTAINING K FACTORS
C     YY           K FACTOR FOR CURRENT TIME INTERVAL
C
      SUBROUTINE RAP011(NEL,NST,TT,I,FLAMB,NT,ICODE)
      DIMENSION NST(50),TT(25),FLAMB(50),Y(25),ICODE(50),XC(6)
      UPDATE(E,T2,T1,F,YY,C)=E+(T2-T1)*F*YY*C
      IF(I-1)9,8,9
    8 T1=0.
      GO TO 10
    9 T1=TT(I-1)
   10 T2=TT(I)
      N=NT-1
      DO 7 J=1,NEL
      IF(ICODE(J)-2)7,1,7
    1 F=FLAMB(J)
      K=5*J+1596
      FETCH(K) (Y(L),L=1,N)
      IF(I-1)12,11,12
   11 YY=Y(1)
      GO TO 13
   12 YY=Y(I-1)
   13 K=4*J-3
      FETCH (K) C,E
      K=K-1
      E=UPDATE(E,T2,T1,F,YY,C)
      RECORD(K) C,E
      KK=NST(J)+1
      DO 6 K=2,KK,3
      L=4*J-3+(K+1)/3
      FETCH(L) (XC(LL),LL=1,6)
      L=L-1
      IF(KK-K-3)2,3,3
    2 LLL=KK
      GO TO 4
    3 LLL=K+2
    4 DO 5 LL=K,LLL
      M=2*(LL-3*((LL-2)/3)-1)-1
      C=XC(M)
      E=XC(M+1)
    5 XC(M+1)=UPDATE(E,T2,T1,F,YY,C)
    6 RECORD (L) (XC(LL),LL=1,6)
    7 CONTINUE
      RETURN
      END
```

Figure B-67. RAP011 Subprogram, Program Listing



ARGUMENTS	
INPUT	OUTPUT
BP	I PAGE
IPAGE	ITAB
ITOTAL	
LS	
LN	
LD	
LM	
T	
NR	
ITAB	
TR	
NUMOR	
NEL	
PMODE	
MCODE	

Figure B-68. RAP012 Subprogram, Logic Flow Chart (1 of 2)

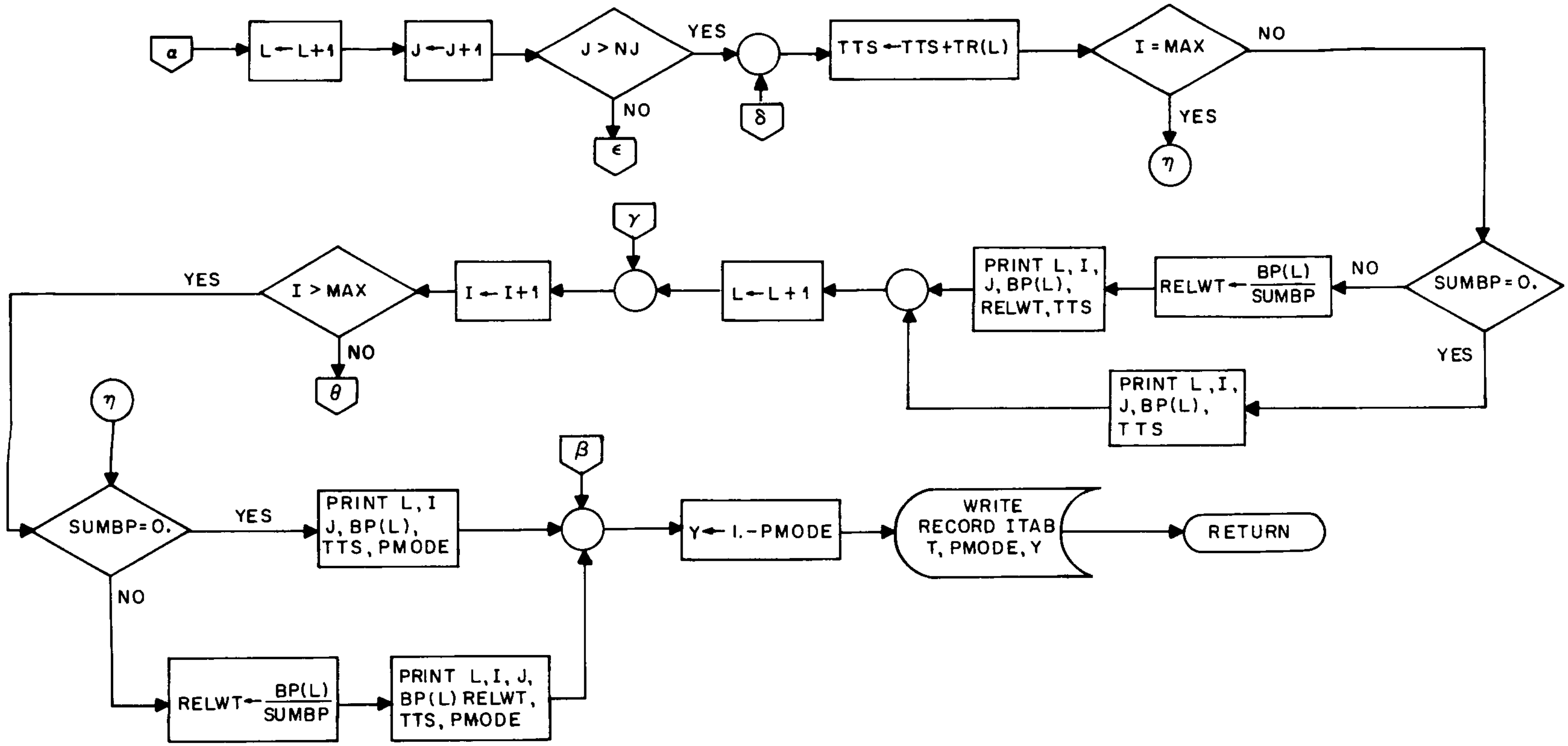


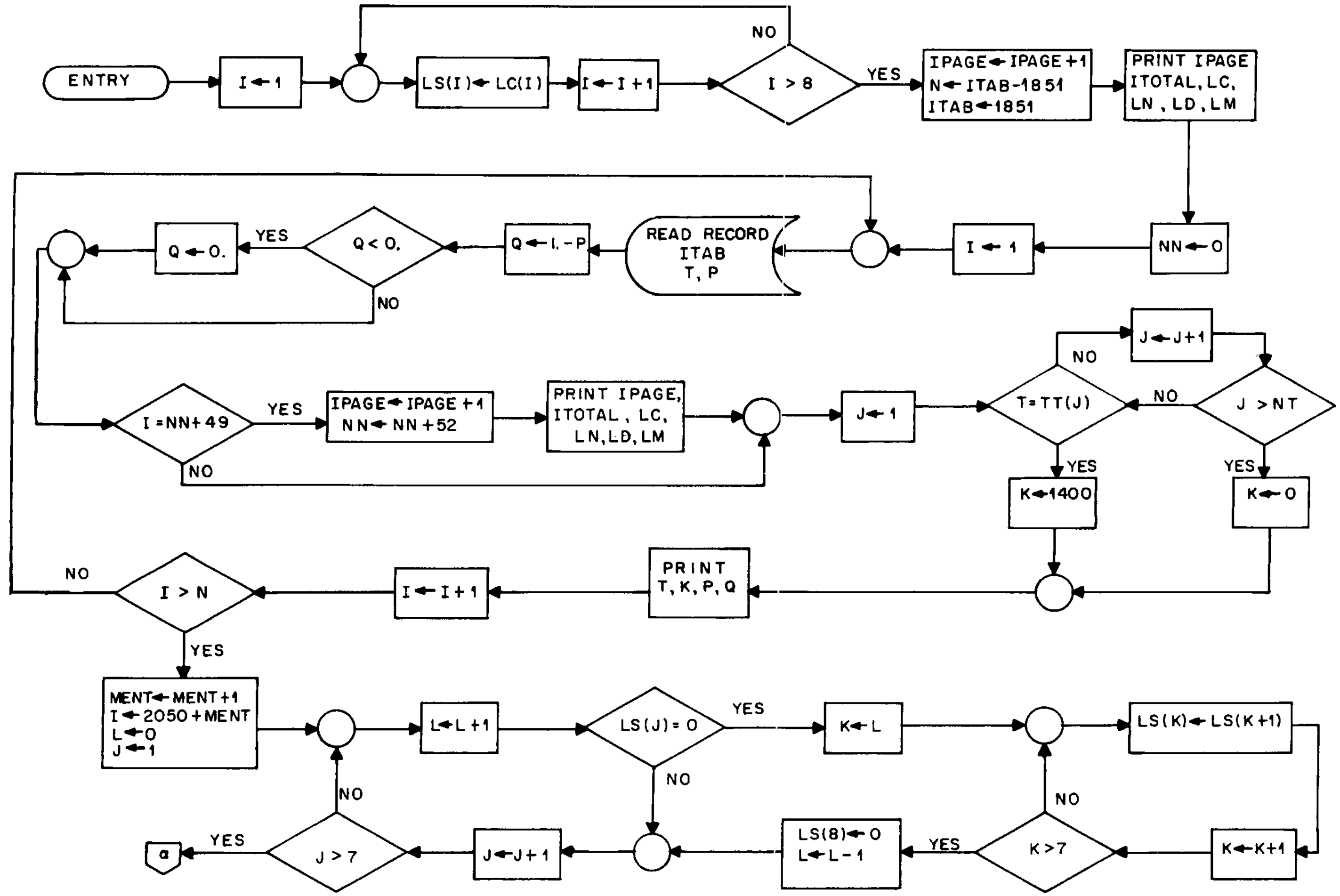
Figure B-68. RAP012 Subprogram, Logic Flow Chart (2 of 2)

```
C
C        RAP012 - - - PRINT OUT DETAILED ANALYSIS PAGE FOR ONE TIME
C
C     BP(100)      BASIC SUBMODE PROBABILITIES
C     I            INDEX
C     IPAGE        PAGE NUMBER
C     ITAB         INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
C     ITOTAL       TOTAL NUMBER OF PAGES
C     J            NUMBER ROW OF ORDER I
C     L            ROW NUMBER
C     LD(3)        DATE OF RUN
C     LM(23)       SYSTEM MODE NAME
C     LN(23)       SYSTEM NAME
C     LS(8)        SYSTEM CODE
C     MAX          LARGEST SUBMODE ORDER FOR CURRENT SMA
C     MCODE        SYSTEM MODE NUMBER
C     NEL          NUMBER OF ELEMENTS IN SYSTEM
C     NJ           INDEX
C     NR           NUMBER OF ROWS IN SYSTEM MODE ARRAY
C     NUMOR(50)    ARRAY CONTAINING NUMBER OF SUBMODES OF EACH ORDER
C     PMODE        PROBABILITY OF SYSTEM MODE OCCURRING
C     RELWT        RELATIVE WEIGHT OF SUBMODE
C     SUMBP        SUM OF THE BASIC PROBABILITIES
C     T            CURRENT TIME
C     TR(100)      SUBMODE PROBABILITY EXCLUSIVE OF OVERLAP
C     TTS          TOTAL ORDER PROBABILITY
C     Y            PROBABILITY OF SYSTEM MODE NON-OCCURENCE
C
      SUBROUTINE RAP012 (BP,IPAGE,ITOTAL,LS,LN,LD,LM,T,NR,ITAB,TR,NUMOR,
     1NEL,PMODE,MCODE)
      DIMENSION BP(100),NUMOR(50),TR(100),LS(8),LN(23),LD(3),LM(23)
  100 FORMAT (1H1,3X,6HRAPID-M1,10X,38H* PROBABILISTIC SYSTEM MODE ANALY
 1001SIS *,5X,4HPAGE,I3,3H OF,I3/1H ,3X,8A1,6X,23A2,6H DATE ,I2,1H/,I2,
 10021H/,I2/1H ,17X,23A2/1H )
  101 FORMAT(78H      --*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*-
     1-*--*--*--*--*--*--*--)
  102 FORMAT(1H ,28X,6HTIME =,F12.5,6H HOURS/1H )
  103 FORMAT (1H ,12X,7HSUBMODE,4X,54HSUBMODE         RELATIVE      TOTAL O
 1031RDER      TOTAL    MODE/1H ,3X,74HSUBMODE    ( O/ N)   PROBABILITY
 1032WEIGHT        PROBABILITY      PROBABILITY/1H )
  104 FORMAT (1H ,4X,I3,5X,1H(,I2,1H/,I2,1H),1X,F11.8,4X,F10.7,2(4X,F11.
 10418))
  105 FORMAT (1H )
  107 FORMAT (1H ,4X,I3,5X,1H(,I2,1H/,I2,1H),1X,F11.8,14X,2(4X,F11.8))
  108 FORMAT(1H ,24X,6HTIME =,F20.5,6H HOURS/1H )
      IF(MCODE)19,16,19
   19 IPAGE=IPAGE+1
      FIND(ITAB)
      PRINT 100 ,IPAGE,ITOTAL,LS,LN,LD,LM
      PRINT 101
      IF(T-100000. )21,20,20
   20 PRINT108,T
      GO TO 22
   21 PRINT102,T
   22 PRINT 103
      SUMBP=0.
      DO 1 J=1,NR
    1 SUMBP=SUMBP+BP(J)
      L=1
      DO 3 I=1,NEL
```

Figure B-69. RAP012 Subprogram, Program Listing (1 of 2)

```
      IF(NUMOR(I))3,3,2
    2 MAX=I
    3 CONTINUE
      DO 12 I=1,MAX
      IF(NUMOR(I))12,12,4
    4 NJ=NUMOR(I)-1
      TTS=0.
      DO 7 J=1,NJ
      IF (NJ)17,18,17
   17 TTS=TTS+TR(L)
      IF(SUMBP)5,6,5
    5 RELWT=BP(L)/SUMBP
      PRINT 104, L,I,J,BP(L),RELWT
      GO TO 7
    6 PRINT 107, L,I,J,BP(L)
    7 L=L+1
   18 TTS=TTS+TR(L)
      IF(I-MAX)8,13,8
    8 IF(SUMBP)9,10,9
    9 RELWT=BP(L)/SUMBP
      PRINT 104, L,I,J,BP(L),RELWT,TTS
      GO TO 11
   10 PRINT 107, L,I,J,BP(L),TTS
   11 PRINT 105
      L=L+1
   12 CONTINUE
   13 IF(SUMBP)14,15,14
   14 RELWT=BP(L)/SUMBP
      PRINT 104, L,I,J,BP(L),RELWT,TTS,PMODE
      GO TO 16
   15 PRINT 107, L,I,J,BP(L),TTS,PMODE
   16 Y=(.99999999-PMODE)+.00000001
      RECORD(ITAB) T,PMODE,Y
      RETURN
      END
```

Figure B-69. RAP012 Subprogram, Program Listing (2 of 2)



ARGUMENTS	
INPUT	OUTPUT
ITAB	MENT
MENT	IPAGE
IPAGE	IAVAIL
ITOTAL	
LS	
LN	
LD	
LM	
IAVAIL	
TT	
NT	
MCODE	

Figure B-70. RAP013 Subprogram, Logic Flow Chart (1 of 2)

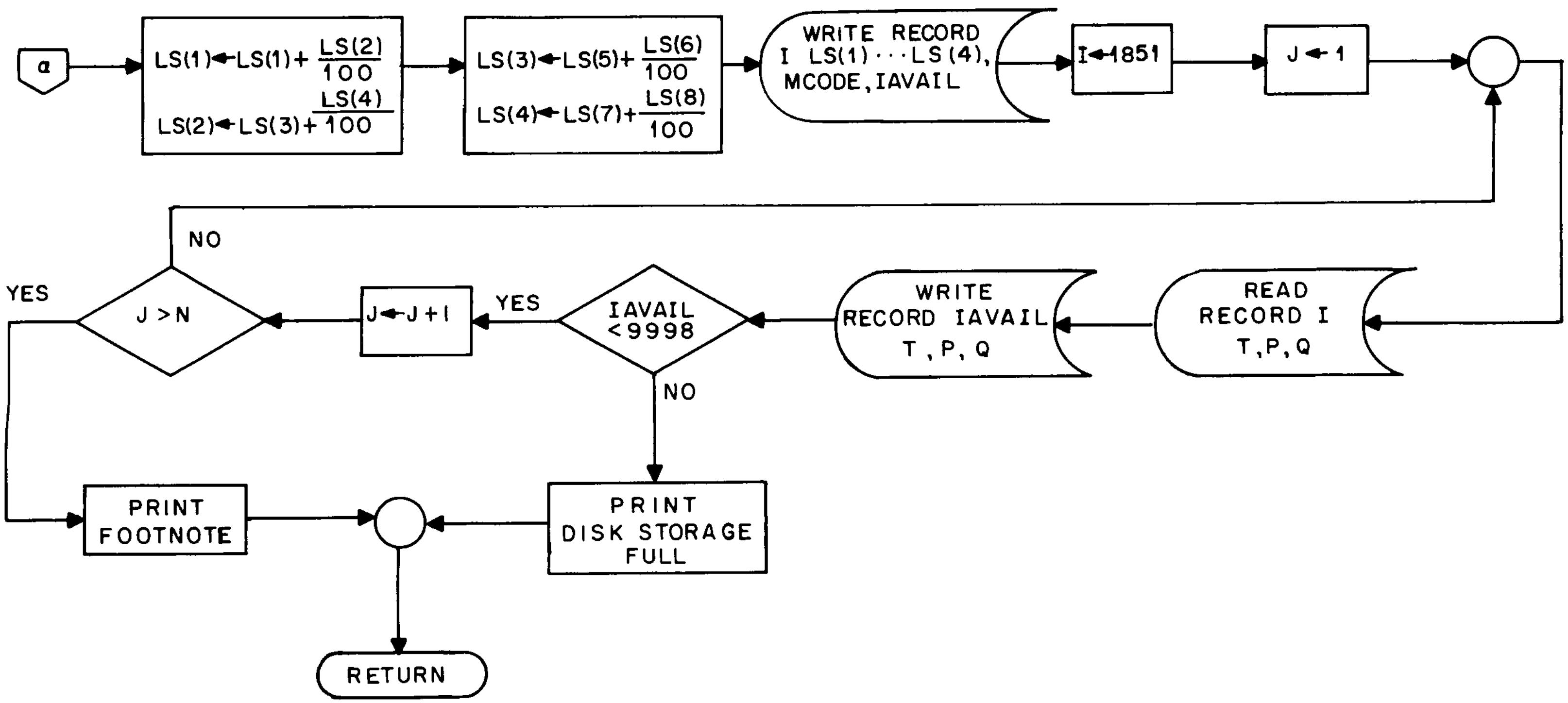


Figure B-70. RAP013 Subprogram, Logic Flow Chart (2 of 2)

```
C
C     RAP013 - - - PRINT OUT SUMMARY TABLE AND RECORD PROBABILITIES
C
C     I            INDEX
C     IAVAIL       CURRENT AVAILABLE RECORD NUMBER
C     IPAGE        PAGE NUMBER
C     ITAB         INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
C     ITOTAL       TOTAL NUMBER OF PAGES
C     J            INDEX
C     K            INDEX
C     L            INDEX
C     LC(8)        SYSTEM CODE
C     LD(3)        DATE OF RUN
C     LM(23)       SYSTEM MODE NAME
C     LN(23)       SYSTEM NAME
C     LS(8)        WORKING ARRAY
C     MCODE        SYSTEM MODE NUMBER
C     MENT         NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
C     N            INDEX
C     NN           INDEX
C     NT           NUMBER OF TRANSITION TIMES
C     P            PROBABILITY OF SYSTEM MODE OCCURENCE
C     Q            PROBABILITY OF SYSTEM MODE NON-OCCURENCE
C     T            CURRENT TIME
C     TT(25)       TRANSITION TIMES
C
      SUBROUTINE RAP013 (ITAB,MENT,IPAGE,ITOTAL,LC,LN,LD,LM,IAVAIL,TT,NT
     1,MCODE)
      DIMENSION LS(8),LN(23),LD(3),LM(23),TT(25),LC(8)
  100 FORMAT (1H1,3X,8HRAPID-M1,10X,38H* PROBABILISTIC SYSTEM MODE ANALY
 1001SIS *,5X,4HPAGE,I3,3H OF,I3/1H ,3X,8A1,6X,23A2,6H DATE ,I2,1H/,I2,
 10021H/,I2/1H ,17X,23A2/1H )
  101 FORMAT(78H      --*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*-
     1-*--*--*--*--*--*--*--)
  102 FORMAT (1H ,28X,24HTOTAL MODE SUMMARY TABLE/1H /1H ,36X,9HOCCURENC
 1021E,6X,11HN-OCCURENCE/1H ,18X,12HTIME (HOURS),5X,11HPROBABILITY,5X,1
 10221HPROBABILITY/1H )
  103 FORMAT (1H ,17X,E13.6,A1,3X,F11.8,5X,F11.8)
  104 FORMAT (1H ,16X,47HSUMMARY OF TOTAL MODE PROBABILITIES (CONTINUED)
 1041/1H /1H ,16X,4HTIME,12X,9HOCCURENCE,9X,13HNON-OCCURENCE)
  105 FORMAT (1H ,29HPROBABILITY DISK STORAGE FULL)
  107 FORMAT (1H0,23X,34HNOTE * INDICATES A TRANSITION TIME)
      DO 12 I=1,8
   12 LS(I)=LC(I)
      IPAGE=IPAGE+1
      N=ITAB-1851
      ITAB=1851
      PRINT 100, IPAGE,ITOTAL,LC,LN,LD,LM
      PRINT 101
      PRINT 102
      NN=0
      DO 3 I=1,N
      FETCH (ITAB) T,P
      Q=(.99999999-P)+.00000001
      IF (Q)13,14,14
   13 Q=0.
   14 IF (I-49-NN)1,2,1
    1 DO 10 J=1,NT
      IF (T-TT(J))10,9,10
    9 K=1400
```

Figure B-71. RAP013 Subprogram, Program Listing (1 of 2)

```
      GO TO 11
10    CONTINUE
      K=0
11    PUNCH 103,T,K,P,Q
      GO TO 3
 2    IPAGE=IPAGE+1
      PRINT 107
      PRINT 100, IPAGE,ITOTAL,LC,LN,LD,LM
      PRINT 101
      PRINT 104
      NN=NN+52
      GO TO 1
 3    CONTINUE
      MENT=MENT+1
      I=2050+MENT
      FIND(I)
      L=0
      DO 6 J=1,7
      L=L+1
      IF (LS(L))6,4,6
 4    DO 5 K=L,7
 5    LS(K)=LS(K+1)
      L=L-1
      LS(8)=0
 6    CONTINUE
      LS(1)=LS(1)+LS(2)/100
      LS(2)=LS(3)+LS(4)/100
      LS(3)=LS(5)+LS(6)/100
      LS(4)=LS(7)+LS(8)/100
      RECORD(I) (LS(J),J=1,4),MCODE,IAVAIL
      I=1851
      DO 8 J=1,N
      FETCH (I) T,P,Q
      RECORD (IAVAIL) T,P,Q
      IF (IAVAIL-9998)8,7,7
 7    PRINT 105
      RETURN
 8    CONTINUE
      PRINT 107
      RETURN
      END
```

Figure B-71. RAP013 Subprogram, Program Listing (2 of 2)

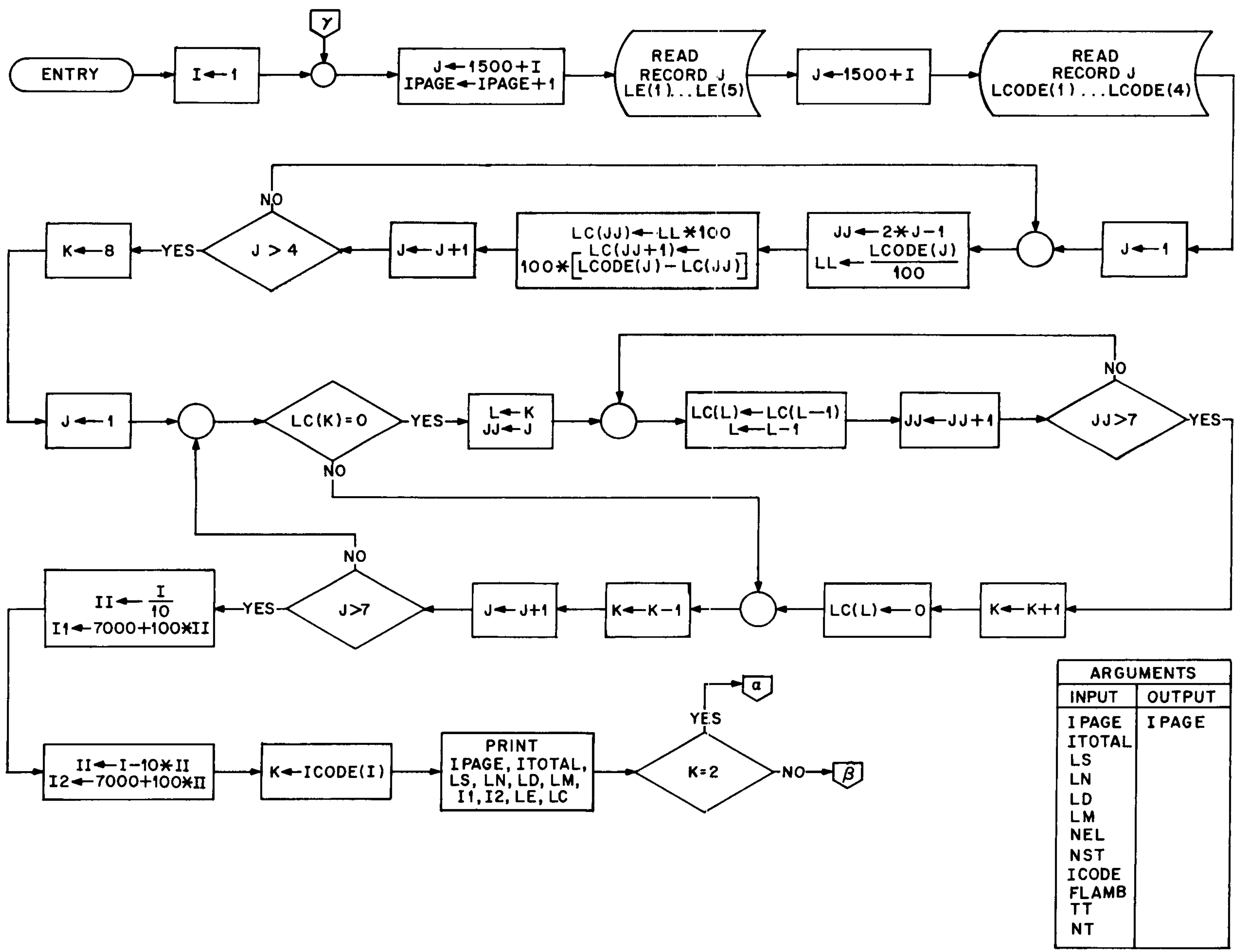


Figure B-72. RAP014 Subprogram, Logic Flow Chart (1 of 2)

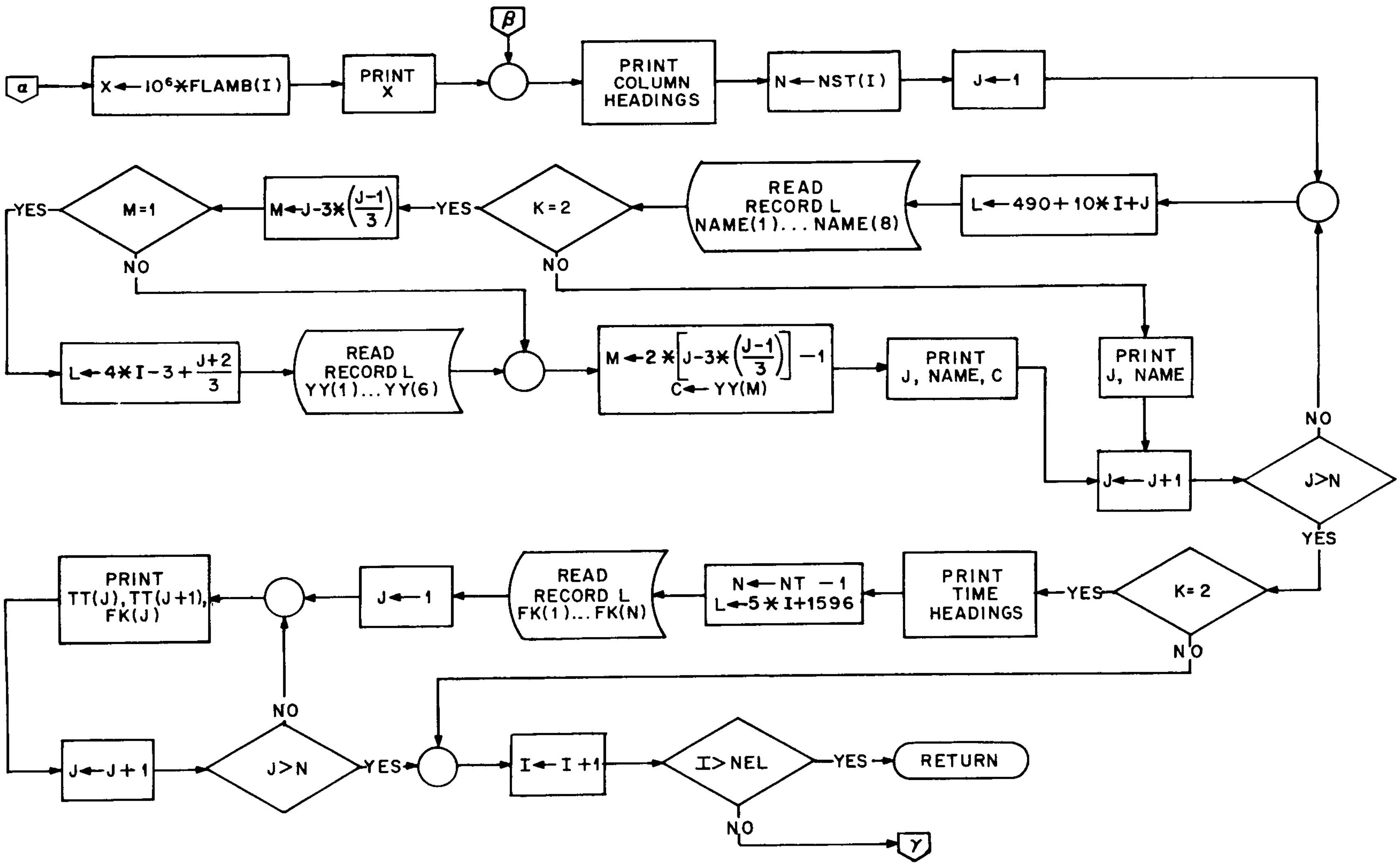


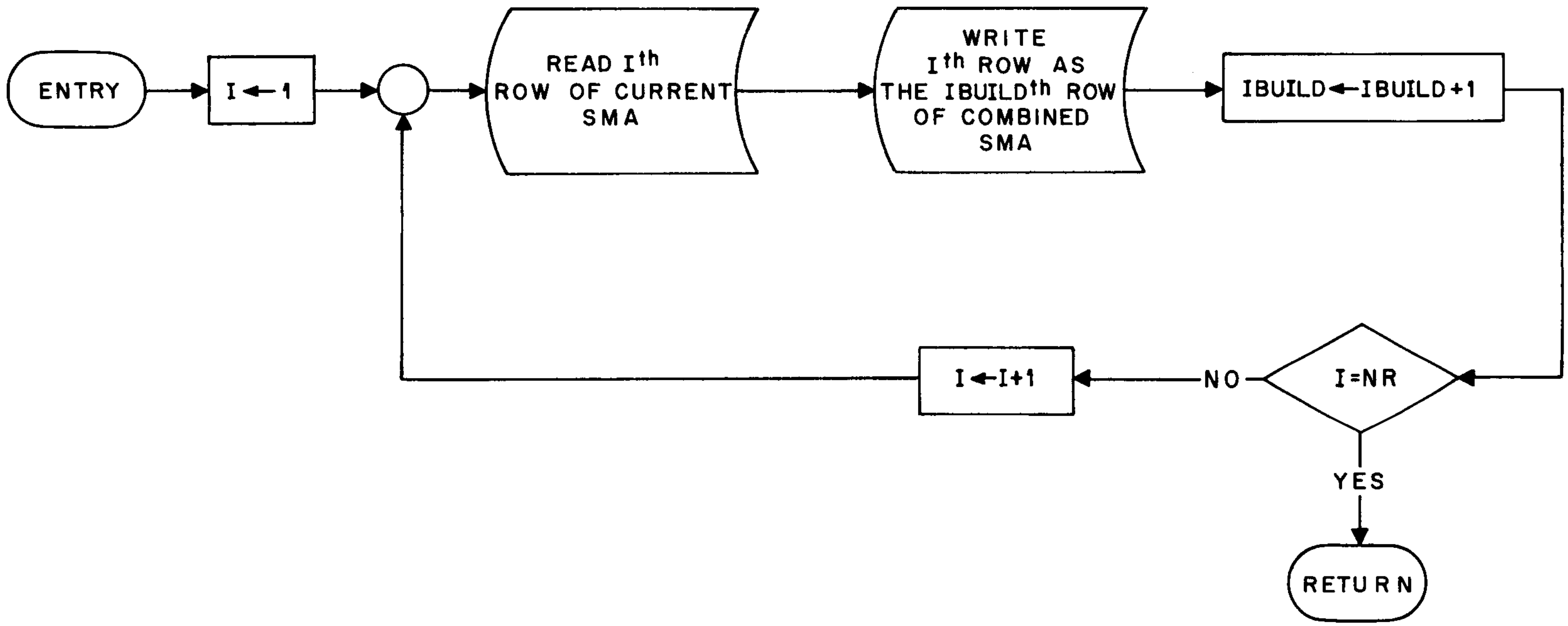
Figure B-72. RAP014 Subprogram, Logic Flow Chart (2 of 2)

```
C
C         RAP014 - - - PRINT OUT ELEMENT DESCRIPTIONS
C
C     C            CONDITIONAL PROBABILITY
C     FK(25)       ARRAY CONTAINING K FACTORS
C     FLAMB(50)    ARRAY CONTAINING FAILURE RATES PER HOUR
C     I            INDEX
C     I1           WORKING VARIABLE
C     I2           WORKING VARIABLE
C     ICODE(50)    ARRAY CONTAINING S CODE
C     II           WORKING VARIABLE
C     IPAGE        PAGE NUMBER
C     ITOTAL       TOTAL NUMBER OF PAGES
C     J            INDEX
C     JJ           INDEX
C     K            INDEX
C     L            INDEX
C     LC(8)        ELEMENT CODE (EXPANDED)
C     LCODE(4)     ELEMENT CODE (COMPRESSED)
C     LD(3)        DATE OF RUN
C     LE(5)        ELEMENT NAME
C     LM(23)       SYSTEM MODE NAME
C     LN(23)       SYSTEM NAME
C     LS(8)        SYSTEM CODE
C     M            INDEX
C     N            INDEX
C     NAME(8)      ELEMENT MODE NAME
C     NEL          NUMBER OF ELEMENTS IN SYSTEM
C     NST(50)      ENTRIES OF THE SYSTEM DEFINITION ARRAY
C     NT           NUMBER OF TRANSITION TIMES
C     TT(25)       TRANSITION TIMES
C     X            FAILURE RATE PER MILLION HOURS
C     YY(6)        WORKING ARRAY
C
      SUBROUTINE RAP014 (IPAGE,ITOTAL,LS,LN,LD,LM,NEL,NST,ICODE,FLAMB,TT
     1,NT)
      DIMENSION NST(50),NAME(8),LCODE(4),ICODE(50),FLAMB(50),TT(25),LS(8
     1),LN(23),LD(3),LM(23),FK(25),LE(5),LC(8)
      DIMENSION YY(6)
  100 FORMAT (1H1,3X,8HRAPID-M1,10X,38H* PROBABILISTIC SYSTEM MODE ANALY
 1001SIS *,5X,4HPAGE,I3,3H OF,I3/1H ,3X,8A1,6X,23A2,6H DATE ,I2,1H/,I2,
 10021H/,I2/1H ,17X,23A2/1H ,3X,74H--*--*--*--*--*--*--*--*--*--*--*
 1003--*--*--*--*--*--*--*--*--*--*--*--*--/1H ,30X,20HELEMENT  DESCRIP
 1004TION/1H ,35X,8HELEMENT-,2A1/1H ,30X,5A2,2X,8A1/1H )
  101 FORMAT (1H ,12X,I2,18X,8A2,14X,F7.4)
  102 FORMAT (1H ,12X,I2,18X,8A2,17X,3H---)
  103 FORMAT (1H0/1H ,20X,22HTIME INTERVAL  (HOURS),10X,8HK FACTOR/1H )
  104 FORMAT (1H ,19X,E10.3,3H TO,E10.3,9X,E9.2)
  105 FORMAT (1H )
  106 FORMAT (1H ,10X,7HELEMENT,44X,11HCONDITIONAL/1H ,8X,11HMODE SYMBOL
 1061,13X,17HELEMENT MODE NAME,12X,11HPROBABILITY/1H0,13X,1H0,25X,3H---
 1062,23X,3H---)
  107 FORMAT(1H ,18X,32HFAILURE RATE PER MILLION HOURS =,F12.5/1H )
  108 FORMAT (1H1)
      DO 6 I=1,NEL
      J=1500+I
      IPAGE=IPAGE+1
      FETCH (J) (LE(K),K=1,5)
      J=1550+I
      FETCH (J)(LCODE(K),K=1,4)
```

Figure B-73. RAP014 Subprogram, Program Listing (1 of 2)

```
      DO 7 J=1,4
      JJ=2*J-1
      LL=LCODE(J)/100
      LC(JJ)=LL*100
    7 LC(JJ+1)=(LCODE(J)-LC(JJ))*100
      K=8
      DO 10 J=1,7
      IF (LC(K))10,8,10
    8 L=K
      DO 9 JJ=J,7
      LC(L)=LC(L-1)
    9 L=L-1
      K=K+1
      LC(L)=0
   10 K=K-1
      II=I/10
      I1=7000+100*II
      II=I-10*II
      I2=7000+100*II
      K=ICODE(I)
      PRINT 100, IPAGE,ITOTAL,LS,LN,LD,LM,I1,I2,LE,LC
      GO TO (12,11,12),K
   11 X=FLAMB(I)*1000000.
      PRINT 107,X
      GO TO 13
   12 PRINT 105
   13 PRINT 106
      N=NST(I)
      DO 18 J=1,N
      L=490+10*I+J
      FETCH(L) (NAME(M),M=1,8)
      GO TO(17,14,17),K
   14 M=J-3*((J-1)/3)
      GO TO(15,16,16),M
   15 L=4*I-3+(J+2)/3
      FETCH(L) (YY(M),M=1,6)
   16 M=2*(J-3*((J-1)/3))-1
      C=YY(M)
      PRINT 101,J,NAME,C
      GO TO 18
   17 PRINT 102,J,NAME
   18 CONTINUE
      GO TO (6,4,6),K
    4 PRINT 103
      N=NT-1
      L=5*I+1596
      FETCH (L)(FK(M),M=1,N)
      DO 5 J=1,N
    5 PRINT 104,TT(J),TT(J+1),FK(J)
    6 CONTINUE
      PRINT 108
      RETURN
      END
```

Figure B-73. RAP014 Subprogram, Program Listing (2 of 2)

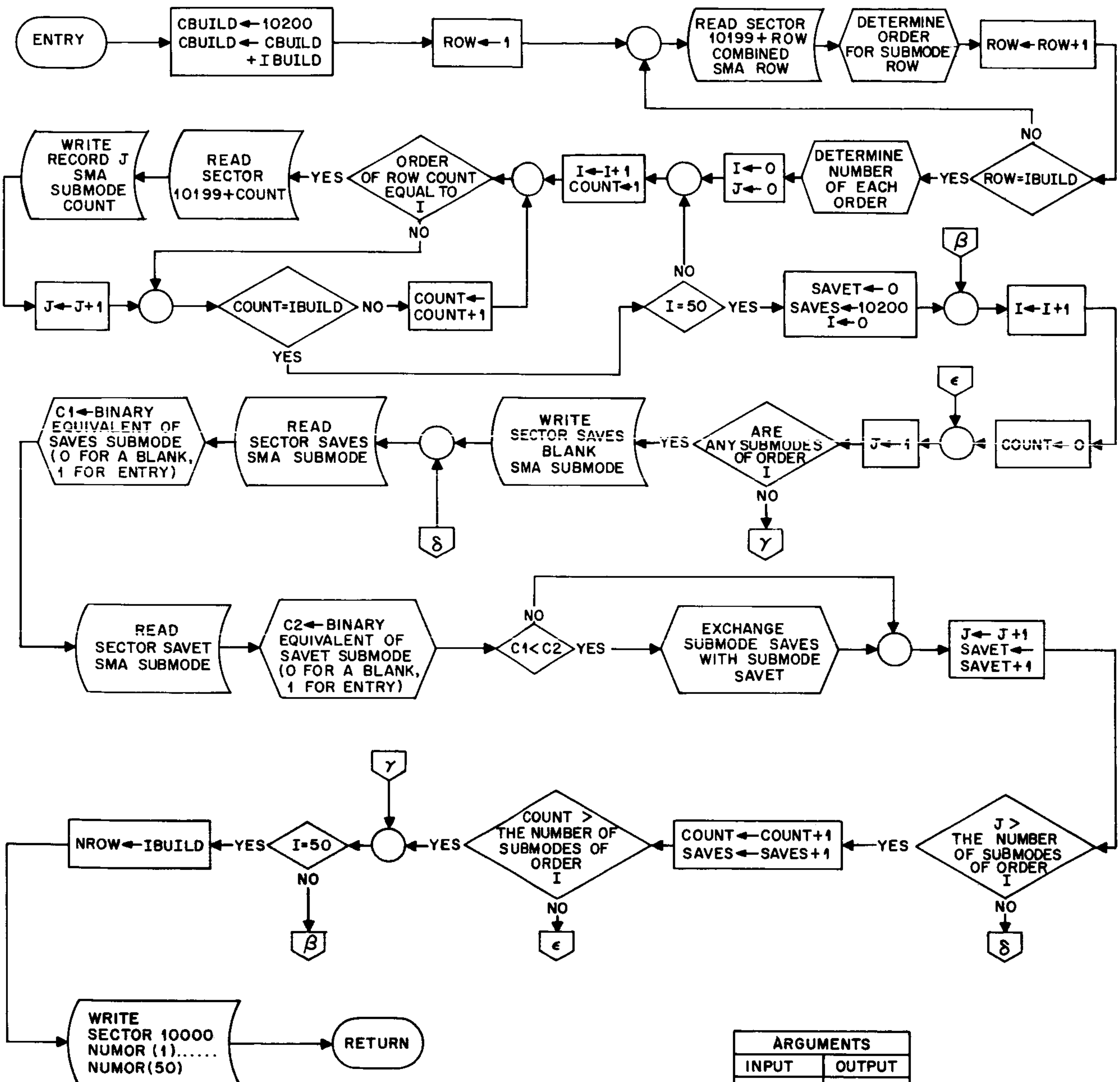


ARGUMENTS	
INPUT	OUTPUT
IBUILD NR	IBUILD

Figure B-74. RAP015 Subprogram, Logic Flow Chart

```
*
*     RAP015 - - - COMBINE SMA'S
*
*     SUBROUTINE LINKAGE
*
S       DS   ,*+101
        DC   6,987898,5-S
        DAC  6,RAP015,7-S
        DVLC22-S,5,LAST,2,8,2,4,5,RAP015-6,5,0,30,0
        DSC  17,0
        DORGS-100
NR      DSA  0
IBUILDDSA  0
        DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*       I              INDEX
*       IBUILD         NUMBER OF ROWS IN COMBINED SMA
*       NR             NUMBER OF ROWS IN CURRENT SMA
*
I       DC   3,0
D1      DDA  ,1,0,1,WORK
IN      DD   ,D1,,,A
D2      DDA  ,1,0,1,WORK
OUT     DD   ,D2,,,A
WORK    DSS  101
*
*     TRANSFER OF ARGUMENTS
*
        DC   5,0
RAP015TFM  TF+6,NR-4
        AM   TF+6,4,10
        AM   RAP015-1,5,10
        TF   CF+11,RAP015-1,11
        BNF  *+36,CF+11
CF      CF   CF+11
        TF   CF+11,CF+11,11
TF      TF   NR,CF+11
        AM   TF+6,1,10
        BNR  RAP015+12,TF+6,11
        AM   RAP015-1,1,10
*
*     START OF SUBROUTINE
*
        TFM  D1+5,10100
        TFM  I,1,9
        TFM  D2+5,10200
        A    D2+5,IBUILD,11
LOOP    GET  IN
        PUT  OUT
        AM   IBUILD,1,610
        C    I,NR,11
        BE   RAP015-1,,6
        AM   I,1,10
        AM   D1+5,1,10
        AM   D2+5,1,10
        B7   LOOP
LAST    DC   2,'
        DENDRAP015
```

Figure B-75. RAP015 Subprogram, Program Listing



ARGUMENTS	
INPUT	OUTPUT
IBUILD	NROW
NEL	

Figure B-76. RAP017 Subprogram, Logic Flow Chart

```
*
*     RAP017 - - - REARRANGE COMBINED SYSTEM MODE ARRAY
*
*     SUBROUTINE LINKAGE
*
S       DS    .*+101
        DC    6.987898.5-S
        DAC 6.RAP017.7-S
        DVLC22-S.5.LAST.2.8.2.4.5.RAP017-6.5.0.30.0
        DSC 17.0.0
        DORGS-100
IBUILDDSA 0
NEL     DSA 0
NROW    DSA 0
        DC    1.'
*
*     PROGRAM CONSTANTS. VARIABLES. AND ARRAYS
*
*       C1            FIELD TO CONTAIN BINARY REPRESENTATION
*       C2            FIELD TO CONTAIN BINARY REPRESENTATION
*       COUNT         COUNTER
*       I             INDEX
*       IBUILD        NUMBER OF ROWS IN COMBINED SMA
*       J             INDEX
*       NEL           NUMBER OF ELEMENTS IN SYSTEM
*       NROW          NUMBER OF ROWS IN SYSTEM MODE ARRAY
*       NUMOR(50)     ARRAY CONTIANING NUMBER OF SUBMODES OF EACH ORDER
*       ORDER(900)    ARRAY CONTAINING THE ORDER OF EACH ROW IN THE
*                     COMBINED SYSTEM MODE ARRAY
*       SAVE          WORK AREA
*       SAVES         SECTOR ADDRESS VARIABLE
*       SAVET         SECTOR ADDRESS VARIABLE
*       WORK          WORK AREA
*       ZERO          CONSTANT TO INITIALIZE WORK AREA
*
D1      DDA   .1.10200.1.WORK
        DC    1.'
DSMA    DD    .D1...A
D2      DDA   .5.0.1.WORK
        DC    1.'
DWORK   DD    .D2...A
D3      DDA   .1.10000.1.NUMOR-1
        DC    1.'
DORDERDD      .D3...A
        DS    1
WORK    DSS 100
COUNT   DS    4
ORDER   DC    2.0
        DS    1798
NUMOR   DC    2.0
        DS    98
I       DS    4
SAVET   DSA 0
SAVES   DSA 0
J       DS    4
C1      DSS 50
        DC    1.'
C2      DSS 50
        DC    1.'
ZERO    DSC 50.0
```

Figure B-77. RAP017 Subprogram, Program Listing (1 of 4)

```
       DC   1,'
SAVE   DSS  100
*
*    TRANSFER OF ARGUMENTS
*
       DC   5,0
RAP017TFM   TF+6,IBUILD-4
       AM   TF+6,4,10
       AM   RAP017-1,5,10
       TF   CF+11,RAP017-1,11
       BNF  *+36,CF+11
CF     CF   CF+11
       TF   CF+11,CF+11,11
TF     TF   IBUILD,CF+11
       AM   TF+6,1,10
       BNR  RAP017+12,TF+6,11
       AM   RAP017-1,2,10
*
*    START OF SUBROUTINE
*
INDEXIBBASBANDA
       BBBSBANDB
       TDM  OUT+11,0,,       SET BAND 0 EXIT
       B7   BANDO
BANDA  TDM  OUT+11,1,,       SET BAND 1 EXIT
       B7   BANDO
BANDB  TDM  OUT+11,2,,       SET BAND 2 EXIT
BANDO  BS   *+12,0,,         SELECT BAND 0 NOW
       TFM  *+18,ZERO
       SF   ZERO
       AM   *-6,1,10
       BNR  *-24,*-18,11
       TFM  *+30,ZERO
       A    *+18,NEL,11
       TD   0,ZERO+50
       TFM  D1+5,10200
       TFM  D2+5,0
       TFM  CBUILD+11,10200
       A    CBUILD+11,IBUILD,11
       TFM  REF+6,ORDER
       TFM  COUNT,0,10
       GET  DSMA
       TFM  LOOP1+11,WORK
LOOP1  BNF  *+20,WORK
       B7   *+20
       AM   COUNT,1,10
       AM   LOOP1+11,1,10
       BNR  LOOP1,LOOP1+11,11
REF    TF   ORDER,COUNT
       TFM  COUNT,0,10
       AM   REF+6,2,10
       AM   D1+5,1,10
CBUILDCM    D1+5,IBUILD
       BNH  LOOP1-36
*
*    INITIALIZE NUMOR TO ZEROES
*
       TFM  COUNT,1,10
       TFM  *+18,NUMOR
INIT   TFM  NUMOR,0,10
```

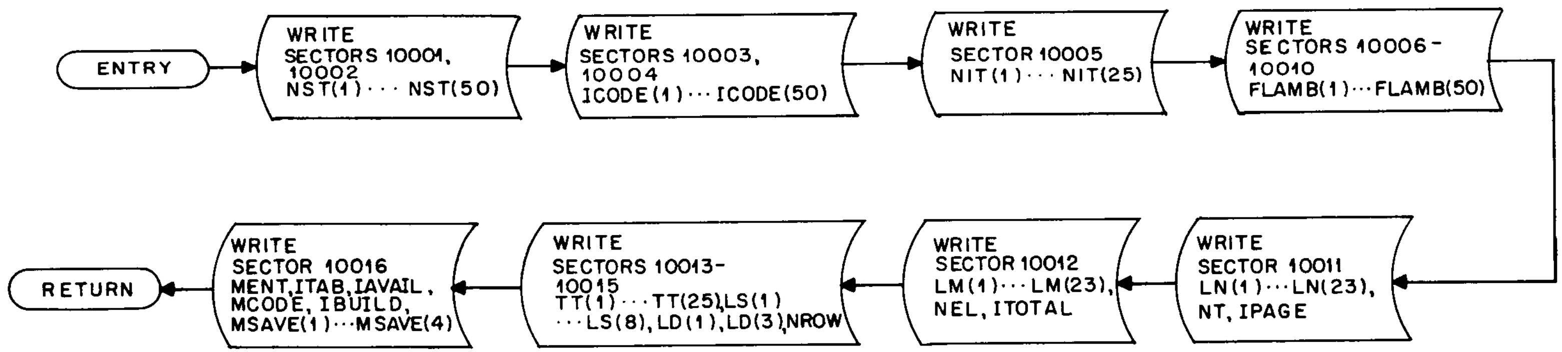
Figure B-77. RAP017 Subprogram, Program Listing (2 of 4)

```
        CM   COUNT,50,10
        BE   *+44
        AM   COUNT,1,10
        AM   INIT+6,2,10
        B7   INIT
*
*     COUNT  THE  ORDERS
*
        TFM  LOOP2+18,ORDER
        TFM  COUNT,0,8
LOOP2   AM   COUNT,1,10
        MM   ORDER,2,10
        SF   95
        AM   99,NUMOR-2
        TF   *+18,99
        AM   NUMOR,1,10
        AM   LOOP2+18,2,10
        C    COUNT,IBUILD,11
        BE   *+20
        B7   LOOP2
*
*     REARRANGE SMA BY ORDER (LOWEST ORDER FIRST)
*
        TFM  D1+5,10200
        TFM  L4+6,ORDER
        TFM  I,1,10
        TFM  COUNT,1,8
L4      C    ORDER,I
        BE   L5
        AM   L4+6,2,10
        C    COUNT,IBUILD,11
        BE   L3
        AM   COUNT,1,10
        AM   D1+5,1,10
        B7   L4
L5      GET  DSMA
        PUT  DWORK
        AM   D2+5,1,10
        B7   L4+24
L3      CM   I,50
        BE   L6
        AM   I,1,10
        TFM  D1+5,10200
        TFM  L4+6,ORDER
        B7   L4-12
L6      TFM  REF1+11,NUMOR-2
        TFM  REF1+59,NUMOR-2
        TFM  I,0,10
        TFM  SAVET,0
        TFM  SAVES,10200
        AM   I,1,10
        AM   REF1+11,2,10
        AM   REF1+59,2,10
L1      TFM  COUNT,1,10
        TF   D1+5,SAVES
        TFM  J,1,10
        TF   D2+5,SAVET
        CM   REF1+11,0,610
        BE   L7
        TR   WORK,ZERO
```

Figure B-77. RAP017 Subprogram, Program Listing (3 of 4)

```
        PUT DSMA
L2      GET DSMA
        TR  C1,WORK
        GET DWORK
        TR  C2,WORK
        BNF *+32,C1
        TDM L2+83,0,6
        B7  *+20
        TDM L2+83,1,6
        AM  L2+83,1,10
        BNR L2+72,L2+83,11
MASK    BNF *+32,C2
        TDM MASK+11,0,6
        B7  *+20
        TDM MASK+11,1,6
        AM  MASK+11,1,10
        BNR MASK,MASK+11,11
        TFM L2+83,C1
        TFM MASK+11,C2
        SF  C1
        SF  C2
        TFM *+54,C1-1
        TFM *+47,C2-1
        A   *+30,NEL,11
        A   *+23,NEL,11
        C   C1,C2
        BNL REF1-24
        GET DSMA
        TR  SAVE,WORK
        GET DWORK
        PUT DSMA
        TR  WORK,SAVE
        PUT DWORK
        AM  J,1,10
        AM  D2+5,1,10
REF1    C   J,NUMOR-2
        BNH L2
        AM  COUNT,1,10
        AM  D1+5,1,10
        C   COUNT,NUMOR-2
        BNH L1+24
L7      CM  I,50,10
        BE  RETURN
        TF  SAVES,D1+5
        TF  SAVET,D2+5
        B7  L6+60
RETURNTF    NROW,IBUILD,611
        PUT DORDER
OUT     BS  RAP017-1,,6
LAST    DC  1,'
        DENDRAP017
```

Figure B-77. RAP017 Subprogram, Program Listing (4 of 4)



ARGUMENTS	
INPUT	OUTPUTS
NST ICODE NIT FLAMB LN LM LS LD NT NEL TT MENT ITAB IAVAIL IPAGE ITOTAL NR MCODE IBUILD MSAVE	

Figure B-78. RAP018 Subprogram, Logic Flow Chart

```
*
*     RAP018 - - - SAVE DATA ON DISK FOR RAPID2
*
*     SUBROUTINE LINKAGE
*
S       DS   ,*+101
        DC   6,987898,5-S
        DAC 6,RAP018,7-S
        DVLC22-S,5,BOTTOM,2,08,2,04,5,RAP018-6,5,0,30,0
        DSC 17,0,0
        DORGS-100
NST     DSA 0
ICODE DSA 0
NIT     DSA 0
FLAMB DSA 0
LN      DSA 0
LM      DSA 0
LS      DSA 0
LD      DSA 0
NT      DSA 0
NEL     DSA 0
TT      DSA 0
MENT    DSA 0
ITAB    DSA 0
IAVAILDSA 0
IPAGE DSA 0
ITOTALDSA 0
NROW    DSA 0
MCODE DSA 0
IBUILDDSA 0
MSAVE DSA 0
        DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*       FLAMB(50)   ARRAY CONTAINING FAILURE RATES PER HOUR
*       HELP        WORK AREA
*       I           INDEX
*       IAVAIL      CURRENT AVAILABLE RECORD NUMBER
*       IBUILD      NUMBER OF ROWS IN COMBINED SMA
*       ICODE(50)   ARRAY CONTAINING S CODE
*       IPAGE       PAGE NUMBER
*       ITAB        INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
*       ITOTAL      TOTAL NUMBER OF PAGES
*       LD(3)       DATE OF RUN
*       LM(23)      SYSTEM MODE NAME
*       LN(23)      SYSTEM NAME
*       LS(8)       SYSTEM CODE
*       MENT        NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
*       MCODE       SYSTEM MODE NUMBER
*       MSAVE(4)    SYSTEM CODE SAVED IN THIS ARRAY
*       NEL         NUMBER OF ELEMENTS IN SYSTEM
*       NIT(25)     NUMBER OF INTERMEDIATE TIMES FOR EACH TRANSITION
*                   INTERVAL
*       NROW        NUMBER OF ROWS IN SYSTEM MODE ARRAY
*       NST(50)     ENTRIES IN THE SYSTEM DEFINITION ARRAY
*       NT          NUMBER OF TRANSITION TIMES
*       TT(25)      TRANSITION TIMES
*
DISK   DDA ,1,10001,0,0
```

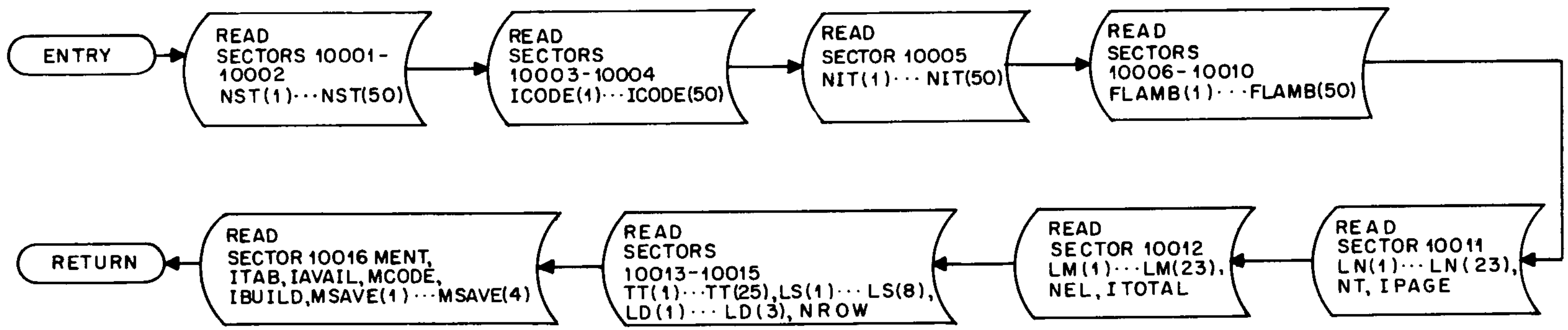
Figure B-79. RAP018 Subprogram, Program Listing (1 of 3)

```
        DC   1,'
OUT     DD   ,DISK,,,A
        DS   1
HELP    DSS  300
I       DS   3
*
*     TRANSFER OF ARGUMENTS
*
        DC   5,0
RAP018TFM    TF+6,NST-4
        AM   TF+6,4,10
        AM   RAP018-1,5,10
        TF   CF+11,RAP018-1,11
        BNF  *+36,CF+11
CF      CF   CF+11
        TF   CF+11,CF+11,11
TF      TF   NST,CF+11
        AM   TF+6,1,10
        BNR  RAP018+12,TF+6,11
        AM   RAP018-1,1,10
*
*     START OF SUBROUTINE
*
        TFM  DISK+5,10001
        TF   DISK+13,NST
        SM   DISK+13,3
        TFM  DISK+8,2,9
        PUT  OUT
        TF   DISK+13,ICODE
        SM   DISK+13,3
        TFM  DISK+5,10003
        PUT  OUT
        TF   DISK+13,NIT
        SM   DISK+13,3
        TFM  DISK+5,10005
        TFM  DISK+8,1,9
        PUT  OUT
        TF   DISK+13,FLAMB
        SM   DISK+13,9
        TFM  DISK+8,5,9
        TFM  DISK+5,10006
        PUT  OUT
        TF   SWITCH+11,LN
        SM   SWITCH+11,3
        TFM  I,1,9
        TFM  *+18,HELP
SWITCHTD     HELP,LN
        CM   I,92,9
        BE   NEXT
        AM   SWITCH+6,1
        AM   SWITCH+11,1
        AM   I,1,9
        B7   SWITCH
NEXT    TFM  DISK+13,HELP
        TFM  DISK+5,10011
        TFM  DISK+8,1,9
        AM   SWITCH+6,4
        TF   SWITCH+6,NT,611
        AM   SWITCH+6,4
        TF   SWITCH+6,IPAGE,611
        PUT  OUT
        TFM  DISK+5,10012
        TF   CLOSE+11,LM
        SM   CLOSE+11,3
        TFM  CLOSE+6,HELP
        TFM  I,1,9
```

Figure B-79. RAP018 Subprogram, Program Listing (2 of 3)

```
CLOSE  TD   HELP,LM
       CM   I,92,9
       BE   EL
       AM   I,1,9
       AM   CLOSE+6,1
       AM   CLOSE+11,1
       B7   CLOSE
EL     AM   CLOSE+6,4
       TF   CLOSE+6,NEL,611
       AM   CLOSE+6,4
       TF   CLOSE+6,ITOTAL,611
       PUT  OUT
       TFM  DISK+5,10013
       TFM  DISK+8,3,9
       TF   TIME+11,TT
       SM   TIME+11,9
       TFM  TIME+6,HELP
       TFM  I,1,9
TIME   TD   HELP,TT
       CM   I,250,9
       BE   *+56
       AM   TIME+6,1
       AM   TIME+11,1
       AM   I,1,9
       B7   TIME
       TFM  I,1,9
       TF   SYMBOL+11,LS
       TF   SYMBOL+6,TIME+6
       AM   SYMBOL+6,4
SYMBOLTF    HELP,LS
       CM   I,8,9
       BE   *+56
       AM   SYMBOL+6,4
       AM   SYMBOL+11,4
       AM   I,1,9
       B7   SYMBOL
       TFM  I,1,9
       TF   DATE+6,SYMBOL+6
       TF   DATE+11,LD
       AM   DATE+6,4
DATE   TF   HELP,LD
       CM   I,3,9
       BE   LAST
       AM   DATE+6,4
       AM   DATE+11,4
       AM   I,1,9
       B7   DATE
LAST   AM   DATE+6,4
       TF   DATE+6,NROW,611
       PUT  OUT
       TFM  DISK+5,10016
       TFM  DISK+8,1,9
       TF   HELP+3,MENT,11
       TF   HELP+7,ITAB,11
       TF   HELP+11,IAVAIL,11
       TF   HELP+15,MCODE,11
       TF   HELP+19,IBUILD,11
       TF   HELP+23,MSAVE,11
       AM   MSAVE,4,10
       TF   HELP+27,MSAVE,11
       AM   MSAVE,4,10
       TF   HELP+31,MSAVE,11
       AM   MSAVE,4,10
       TF   HELP+35,MSAVE,11
       PUT  OUT
       B7   RAP018-1,,6
BOTTOMDC    2, '
       DENDRAP018
```

Figure B-79. RAP018 Subprogram, Program Listing (3 of 3)



ARGUMENTS	
INPUT	OUTPUT
	NST ICODE NIT FLAMB LN LM LS LD NT NEL TT MENT ITAB IAVAIL IPAGE ITOTAL NR MCODE IBUILD NUMOR MSAVE

Figure B-80. RAP019 Subprogram, Logic Flow Chart

```
*
*     RAP019 - - - RETRIEVE DATA FROM RAPID1
*
*     SUBROUTINE LINKAGE
*
S      DS   ,*+101
       DC   6,987898,5-S
       DAC  6,RAP019,7-S
       DVLC22-S,5,LAST,2,08,2,04,5,RAP019-6,5,0,30,0
       DSC  17,0,0
       DORGS-100
NST    DSA  0
ICODE  DSA  0
NIT    DSA  0
FLAMB  DSA  0
LN     DSA  0
LM     DSA  0
LS     DSA  0
LD     DSA  0
NT     DSA  0
NEL    DSA  0
TT     DSA  0
MENT   DSA  0
ITAB   DSA  0
IAVAILDSA  0
IPAGE  DSA  0
ITOTALDSA  0
NROW   DSA  0
MCODE  DSA  0
IBUILDDSA  0
NUMOR  DSA  0
MSAVE  DSA  0
       DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*      FLAMB(50)    ARRAY CONTAINING FAILURE RATES PER HOUR
*      HELP         WORK AREA
*      I            INDEX
*      IAVAIL       CURRENT AVAILABLE RECORD NUMBER
*      IBUILD       NUMBER OF ROWS IN COMBINED SMA
*      ICODE(50)    ARRAY CONTAINING S CODE
*      IPAGE        PAGE NUMBER
*      ITAB         INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
*      ITOTAL       TOTAL NUMBER OF PAGES
*      LD(3)        DATE OF RUN
*      LM(23)       SYSTEM MODE NAME
*      LN(23)       SYSTEM NAME
*      LS(8)        SYSTEM CODE
*      MENT         NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
*      MCODE        SYSTEM MODE NUMBER
*      MSAVE(4)     SYSTEM CODE SAVED IN THIS ARRAY
*      NEL          NUMBER OF ELEMENTS IN SYSTEM
*      NIT(25)      NUMBER OF INTERMEDIATE TIMES FOR EACH TRANSITION
*                   INTERVAL
*      NROW         NUMBER OF ROWS IN SYSTEM MODE ARRAY
*      NST(50)      ENTRIES OF THE SYSTEM DEFINITION ARRAY
*      NT           NUMBER OF TRANSITION TIMES
*      TT(25)       TRANSITION TIMES
*
```

Figure B-81. RAP019 Subprogram, Program Listing (1 of 4)

```
DISK   DDA  ,1,10001,0,0
       DC   1,'
IN     DD   ,DISK,,,A
       DS   1
HELP   DSS  300
I      DS   3
*
*     TRANSFER OF ARGUMENTS
*
       DC   5,0
RAP019TFM   TF+6,NST-4
       AM   TF+6,4,10
       AM   RAP019-1,5,10
       TF   CF+11,RAP019-1,11
       BNF  *+36,CF+11
CF     CF   CF+11
       TF   CF+11,CF+11,11
TF     TF   NST,CF+11
       AM   TF+6,1,10
       BNR  RAP019+12,TF+6,11
       AM   RAP019-1,2,10
*
*     START OF SUBROUTINE
*
INDEXIBBASBANDA
       BBBSBANDB
       TDM  GOODBY+11,0,,     SET BAND 0 EXIT
       B7   BANDO
BANDA  TDM  GOODBY+11,1,,     SET BAND 1 EXIT
       B7   BANDO
BANDB  TDM  GOODBY+11,2,,     SET BAND 2 EXIT
BANDO  BS   *+12,0,,          SELECT BAND 0 NOW
       TFM  DISK+5,10001
       TF   DISK+13,NST
       SM   DISK+13,3
       TFM  DISK+8,2,9
       GET  IN
       TF   DISK+13,ICODE
       SM   DISK+13,3
       AM   DISK+5,2
       GET  IN
       TF   DISK+13,NIT
       SM   DISK+13,3
       AM   DISK+5,2
       TFM  DISK+8,1,9
       GET  IN
       TF   DISK+13,FLAMB
       SM   DISK+13,9
       TFM  DISK+8,5,9
       AM   DISK+5,1
       GET  IN
       TFM  DISK+13,HELP
       AM   DISK+5,5
       TFM  DISK+8,1,9
       GET  IN
       TF   SWITCH+6,LN
       SM   SWITCH+6,3
       TFM  I,1,9
       TFM  SWITCH+11,HELP
SWITCHTD    LN,HELP
```

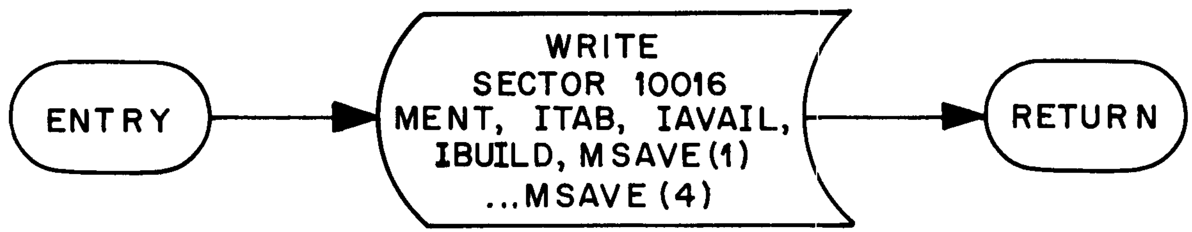
Figure B-81. RAP019 Subprogram, Program Listing (2 of 4)

```
        CM    I,92,9
        BE    NEXT
        AM    SWITCH+6,1
        AM    SWITCH+11,1
        AM    I,1,9
        B7    SWITCH
NEXT    AM    SWITCH+11,4
        TF    NT,SWITCH+11,611
        AM    DISK+5,1
        AM    SWITCH+11,4
        TF    IPAGE,SWITCH+11,611
        GET   IN
        TF    CLOSE+6,LM
        SM    CLOSE+6,3
        TFM   CLOSE+11,HELP
        TFM   I,1,9
CLOSE   TD    LM,HELP
        CM    I,92,9
        BE    EL
        AM    I,1,9
        AM    CLOSE+6,1
        AM    CLOSE+11,1
        B7    CLOSE
EL      AM    CLOSE+11,4
        TF    NEL,CLOSE+11,611
        AM    CLOSE+11,4
        TF    ITOTAL,CLOSE+11,611
        AM    DISK+5,1
        TFM   DISK+8,3,9
        GET   IN
        TF    TIME+6,TT
        SM    TIME+6,9
        TFM   TIME+11,HELP
        TFM   I,1,9
TIME    TD    TT,HELP
        CM    I,250,9
        BE    *+56
        AM    TIME+6,1
        AM    TIME+11,1
        AM    I,1,9
        B7    TIME
        TFM   I,1,9
        TF    SYMBOL+6,LS
        TF    SYMBOL+11,TIME+11
        AM    SYMBOL+11,4
SYMBOLTF      LS,HELP
        CM    I,8,9
        BE    *+56
        AM    SYMBOL+6,4
        AM    SYMBOL+11,4
        AM    I,1,9
        B7    SYMBOL
        TFM   I,1,9
        TF    DATE+11,SYMBOL+11
        TF    DATE+6,LD
        AM    DATE+11,4
DATE    TF    LD,HELP
        CM    I,3,9
        BE    GONE
        AM    DATE+6,4
```

Figure B-81. RAP019 Subprogram, Program Listing (3 of 4)

```
        AM   DATE+11,4
        AM   I,1,9
        B7   DATE
GONE    AM   DATE+11,4
        TF   NROW,DATE+11,611
        TFM  DISK+5,10016
        TFM  DISK+8,1,9
        GET  IN.
        TF   MENT,HELP+3,6
        TF   ITAB,HELP+7,6
        TF   IAVAIL,HELP+11,6
        TF   MCODE,HELP+15,6
        TF   IBUILD,HELP+19,6
        TF   MSAVE,HELP+23,6
        AM   MSAVE,4,10
        TF   MSAVE,HELP+27,6
        AM   MSAVE,4,10
        TF   MSAVE,HELP+31,6
        AM   MSAVE,4,10
        TF   MSAVE,HELP+35,6
        TFM  DISK+5,10000
        GET  IN
        TFM  I,1,9
        TFM  ORDER+11,HELP+1
        TF   ORDER+6,NUMOR
        TF   ORDER+18,NUMOR
        SM   ORDER+18,2
ORDER   TF   NUMOR,HELP+1
        TFM  NUMOR-2,0,10
        AM   ORDER+18,1
        CF   ORDER+18,,6
        CM   I,50,9
        BE   GOODBY
        AM   ORDER+6,4
        AM   ORDER+11,2
        AM   ORDER+18,3
        AM   I,1,9
        B7   ORDER
GOODBYBS     RAP019-1,,6
LAST    DC   1,'
        DENDRAP019
```

Figure B-81. RAP019 Subprogram, Program Listing (4 of 4)

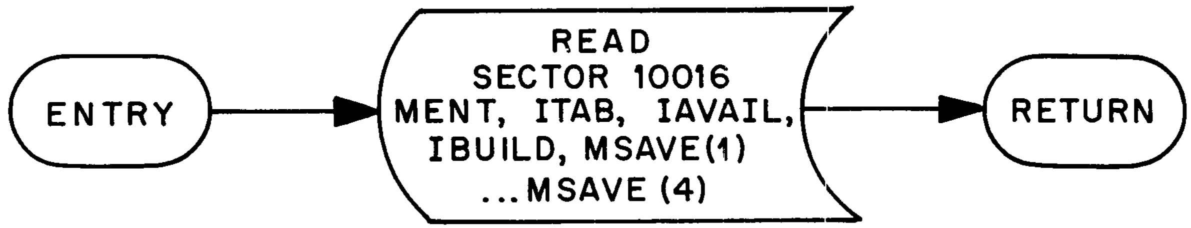


ARGUMENTS	
INPUT	OUTPUT
MENT ITAB IAVAIL IBUILD MSAVE	

Figure B-82. RAP020 Subprogram, Logic Flow Chart

```
*
*     RAP020 - - - SAVE DATA FOR RAPID1
*
*     SUBROUTINE LINKAGE
*
S       DS   ,*+101
        DC   6,987898,5-S
        DAC  6,RAP020,7-S
        DVLC22-S,5,LAST,2,8,2,4,5,RAP020-6,5,0,30,0
        DSC  17,0,0
        DORGS-100
MENT    DSA  0
ITAB    DSA  0
IAVAILDSA    0
IBUILDDSA    0
MSAVE   DSA  0
        DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*       IAVAIL        CURRENT AVAILABLE RECORD NUMBER
*       IBUILD        NUMBER OF ROWS IN COMBINED SMA
*       ITAB          INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
*       MENT          NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
*       MSAVE(4)      SYSTEM CODE SAVED IN THIS ARRAY
*       WORK          WORK AREA
*
WORK    DSS  100
D1      DDA  ,1,10016,1,WORK
        DC   1,'
DISK    DD   ,D1,,,A
*
*     TRANSFER OF ARGUMENTS
*
        DC   5,0
RAP020TFM    TF+6,MENT-4
        AM   TF+6,4,10
        AM   RAP020-1,5,10
        TF   CF+11,RAP020-1,11
        BNF  *+36,CF+11
CF      CF   CF+11
        TF   CF+11,CF+11,11
TF      TF   MENT,CF+11
        AM   TF+6,1,10
        BNR  RAP020+12,TF+6,11
        AM   RAP020-1,2,10
*
*     START OF SUBROUTINE
*
        TF   WORK+3,MENT,11
        TF   WORK+7,ITAB,11
        TF   WORK+11,IAVAIL,11
        TF   WORK+19,IBUILD,11
        TF   WORK+23,MSAVE,11
        AM   MSAVE,4,10
        TF   WORK+27,MSAVE,11
        AM   MSAVE,4,10
        TF   WORK+31,MSAVE,11
        AM   MSAVE,4,10
        TF   WORK+35,MSAVE,11
        PUT  DISK
        B7   RAP020-1,,6
LAST    DC   2, '
        DENDRAP020
```

Figure B-83. RAP020 Subprogram, Program Listing



ARGUMENT	
INPUT	OUTPUT
	MENT ITAB IAVAIL IBUILD MSAVE

Figure B-84. RAP021 Subprogram, Logic Flow Chart

```
*
*     RAP021 - - - RETRIEVE DATA FROM RAPID2 OR RAP022
*
*     SUBROUTINE LINKAGE
*
S       DS   ,*+101
        DC   6,987898,5-S
        DAC  6,RAP021,7-S
        DVLC22-S,5,LAST,2,08,2,04,5,RAP021-6,5,0,30,0
        DSC  17,0,0
        DORGS-100
MENT    DSA  0
ITAB    DSA  0
IAVAILDSA  0
IBUILDDSA  0
MSAVE DSA  0
        DC   1,'
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*       IAVAIL        CURRENT AVAILABLE RECORD NUMBER
*       IBUILD        NUMBER OF ROWS IN COMBINED SMA
*       ITAB          INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
*       MENT          NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
*       MSAVE(4)      SYSTEM CODE SAVED IN THIS ARRAY
*       WORK          WORK AREA
*
WORK    DSS  100
D1      DDA  ,1,10016,1,WORK
        DC   1,'
DISK    DD   ,D1,,,A
*
*     TRANSFER OF ARGUMENTS
*
        DC   5,0
RAP021TFM  TF+6,MENT-4
        AM   TF+6,4,10
        AM   RAP021-1,5,10
        TF   CF+11,RAP021-1,11
        BNF  *+36,CF+11
CF      CF   CF+11
        TF   CF+11,CF+11,11
TF      TF   MENT,CF+11
        AM   TF+6,1,10
        BNR  RAP021+12,TF+6,11
        AM   RAP021-1,2,10
*
*     START OF SUBROUTINE
*
        GET  DISK
        TF   MENT,WORK+3,6
        TF   ITAB,WORK+7,6
        TF   IAVAIL,WORK+11,6
        TF   IBUILD,WORK+19,6
        TF   MSAVE,WORK+23,6
        AM   MSAVE,4,10
        TF   MSAVE,WORK+27,6
        AM   MSAVE,4,10
        TF   MSAVE,WORK+31,6
        AM   MSAVE,4,10
        TF   MSAVE,WORK+35,6
        B7   RAP021-1,,6
LAST    DC   2, '
        DENDRAP021
```

Figure B-85. RAP021 Subprogram, Program Listing

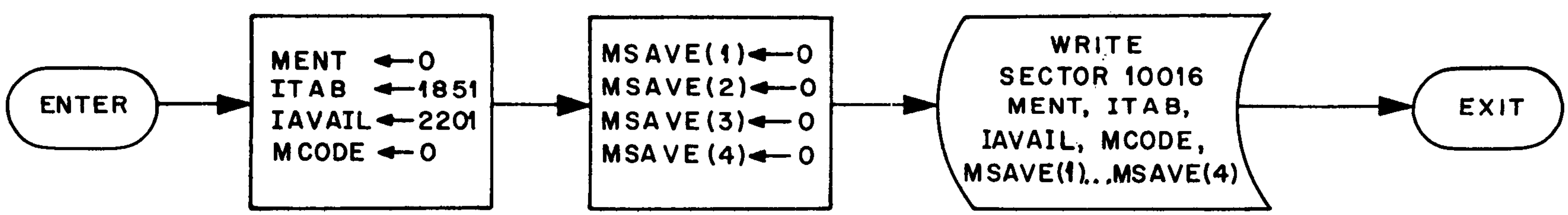


Figure B-86. RAP022 Subprogram, Logic Flow Chart

```
*
*     RAP022 - - - INITIALIZES INDICATORS AND TABLES FOR RAPID
*
       DORG2402
*
*     PROGRAM CONSTANTS, VARIABLES, AND ARRAYS
*
*      IAVAIL       CURRENT AVAILABLE RECORD NUMBER
*      IBUILD       NUMBER OF ROWS IN COMBINED SMA
*      ITAB         INDEX TO NEXT ENTRY IN PROBABILITY SUMMARY TABLE
*      MENT         NUMBER OF ENTRIES IN THE SYSTEM CODE TABLE
*      MSAVE(4)     SYSTEM CODE SAVED IN THIS ARRAY
*
MENT   DC   4,0
ITAB   DC   4,1851
IAVAILDC   4,2201
MCODE  DC   4,0
IBUILDDC   4,0
MSAVE1DC   4,0
MSAVE2DC   4,0
MSAVE3DC   4,0
MSAVE4DC   4,0
       DS   74
D1     DDA  ,1,10016,1,MENT-3
       DC   1,'
DISK   DD   ,D1,,,A
*
*     START OF PROGRAM
*
RAP022SEEKDISK
       PUT DISK
       CALLEXIT
       DENDRAP022
```

Figure B-87. RAP022 Subprogram, Program Listing

APPENDIX C

ECONOMICS SIMULATOR

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SUMMARY

This appendix outlines the framework required to formulate a system economics model. Available cost data are used to generate cost functions which express element cost as a function of operational parameters such as reliability and performance. The Economics Simulator formulates the economics model for the system by summing the individual element cost functions.

1 INTRODUCTION

The PRESTO concept has been advanced as a means for enhancing trade-off studies in the selection and design of modern complex systems. Economics simulation plays an important role in this area, since system cost has a strong influence on the trade-off decisions.

An economics model is required to express the system cost as a function of operational parameters of the elements in the system. The simulation method described here uses available cost data in the generation of element cost functions which are used to construct the economics model of the system.

The basic approach is illustrated by the logic diagram presented in Figure C-1.

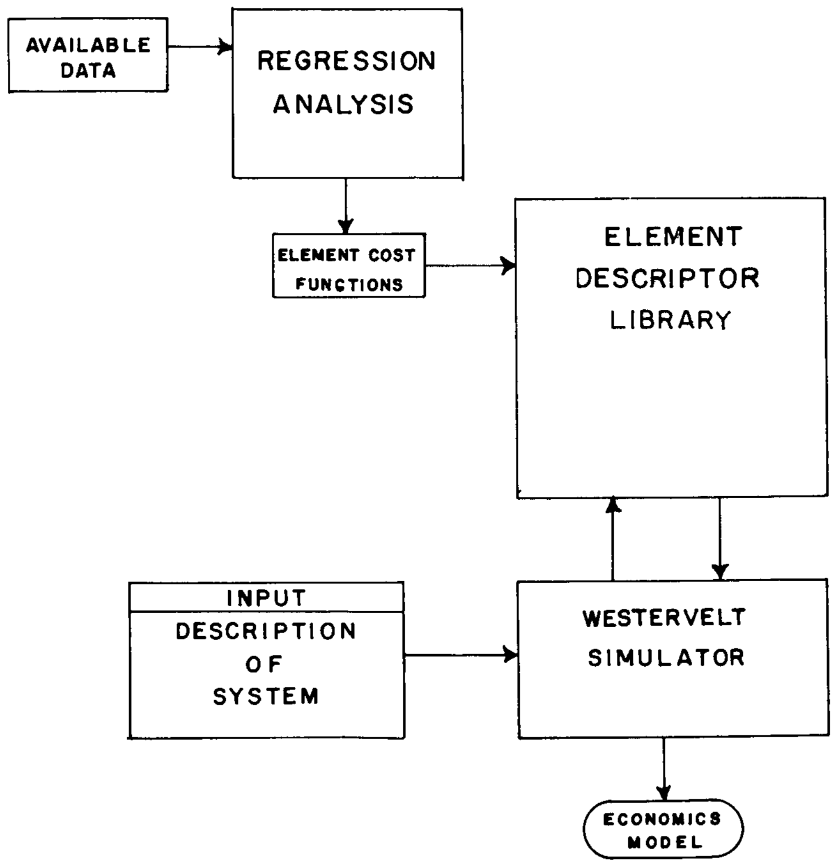


Figure C-1. System Economics Simulation Logic Diagram

2 ELEMENT COST FUNCTION

In order to obtain an element cost function, a definition of the element parameters related to cost is required. While many areas affect element cost, this discussion is limited to the following: (1) the element reliability, which is normally described by the element failure rate parameter, and (2) the element performance, involving those parameters which describe how well the element accomplishes its objectives in the system.

Table C-I illustrates the type of information required to generate the desired element cost function for a gyro. The gyro cost is associated with various levels of failure rate and typical performance parameters including bias drift, random drift, mass unbalance, and anisoelasticity.

It is evident from Table C-I that the combination of parameters associated with a given cost is not unique. As the number of parameters (independent variables) increases, the number of combinations yielding a specific cost (dependent variable) grows rapidly.

The problem of generating element cost functions is, therefore, a multidimensional one which requires techniques capable of investigating the interactions among the independent variables. Analysis using regression techniques may be performed on relatively large quantities of available data of the type shown in Table C-I, and a functional relationship for cost in terms of the independent parameters can be established. This relationship will be a mathematical expression involving the significant element parameters and their interactions. As such, these functions cannot be plotted in a two-axis graph except in a qualitative sense which relates a single lumped variable called "performance" to the element cost. Figure C-2 attempts to express this relationship.

The functions resulting from the regression analysis should, prior to their application to a system economics simulation, be analyzed to ascertain reasonable correlation with reality.

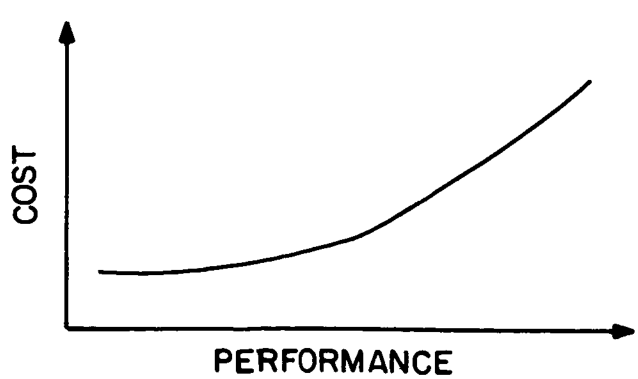


Figure C-2. Cost Vs. Performance Curve

Table C-I

Data Set Number	Cost (Dollars)	Failure Rate (λ) (Per Million Hours)	Bias Drift (deg/hr)	Random Drift (deg/hr)	Mass Unbalance Drift (deg/hr/g)	Anisoelastic Drift (deg/hr/g^2)
1	$ 2,000.00	2000	8.0	5.0	8.0	1.0
2	3,750.00	1750	6.0	4.0	6.0	0.5
3	5,000.00	1750	2.0	4.0	1.0	0.2
4	6,500.00	1600	1.0	1.0	1.0	0.1
5	12,000.00	750	0.75	0.05	0.5	0.05
⋮	⋮	⋮	⋮	⋮	⋮	⋮

3 SYSTEM ECONOMICS MODEL

A suggested procedure for the generation of the system economics model utilizes the Westervelt Simulator (Appendix A). This simulator has the intended capability of generating a digital computer program which will explicitly relate the total cost of the system to element reliability and performance parameters through summation of the element cost functions.

In order to implement the Westervelt Simulator as an economics simulator, the system must be defined with its attachments. An example of a three-element system is shown in Figure C-3. This definition is consistent with the conventions delineated in Appendix A, with the one addition that a system element called cost is attached to all other elements in the system.

Each Element Description must contain a Statement Collection defining the element cost function. The Element Description for cost contains one Statement Collection defining the system cost as the sum of the cost functions of the attached elements.

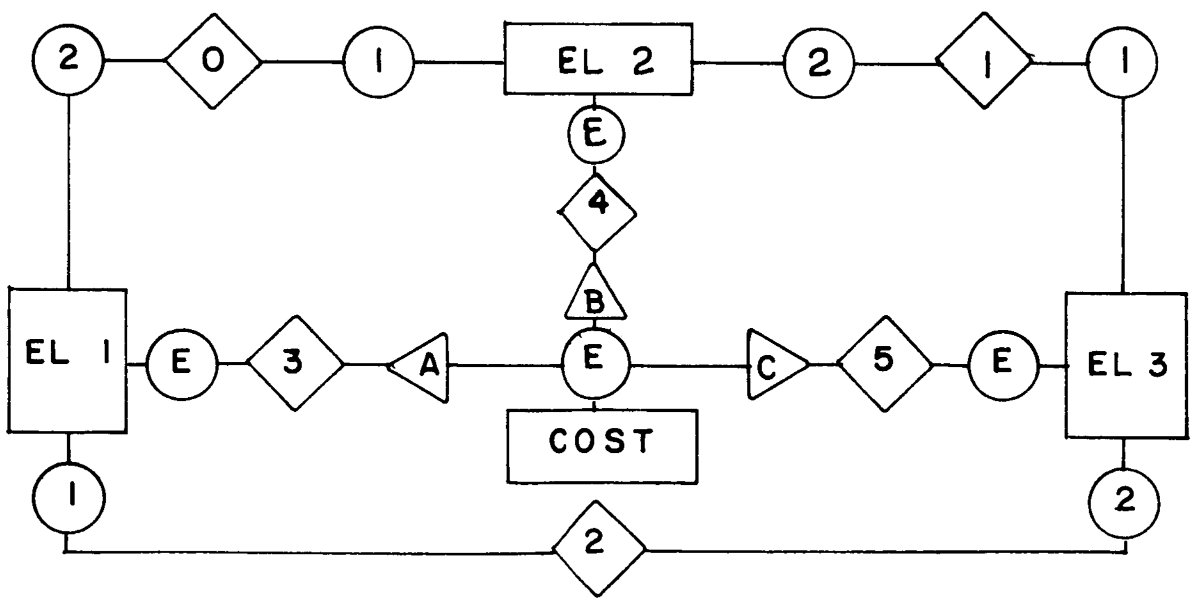


Figure C-3. Economics Simulation Format of Three-Element System

4 EXAMPLE

An example is presented to illustrate the manner in which the Westervelt Simulator might be used to generate the economics model for the system shown in Figure C-4. The required inputs are shown as they would appear for the simulator, but since the simulator is not completely functional, no output is available.

The system diagram in economics simulator format appears in Figure C-5.

4.1 SOURCE PROGRAM

Adource program (Figure C-6) which includes the following information, is required as input to the economics simulator.

- System Description (Connection Statements)
- Input Parameters
- Desired Results
- New Element Tape

Instructions for preparing the above information for the simulator are given in Appendix G.

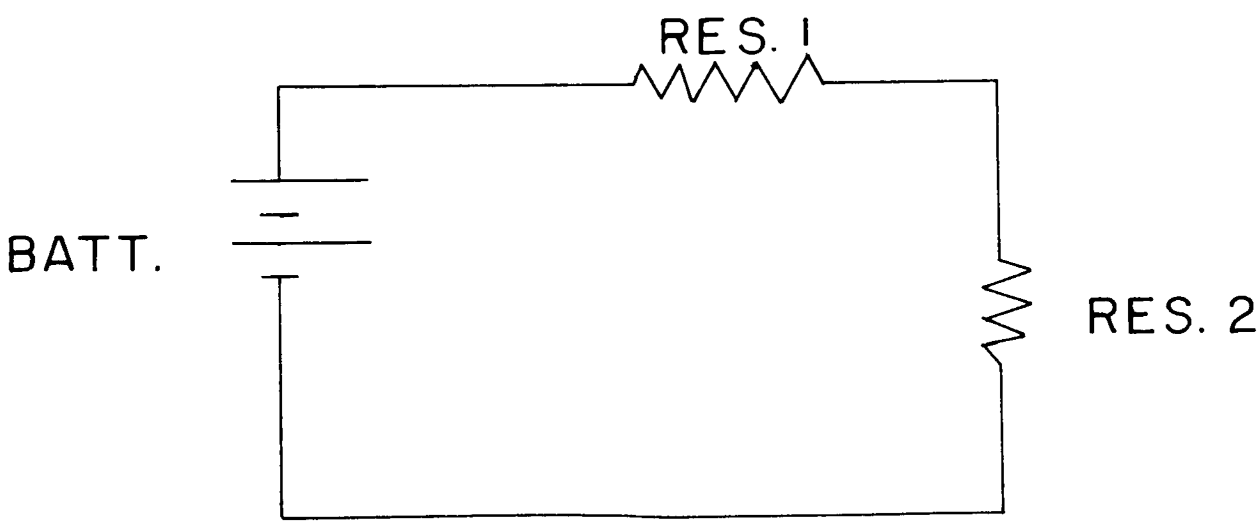


Figure C-4. Example System

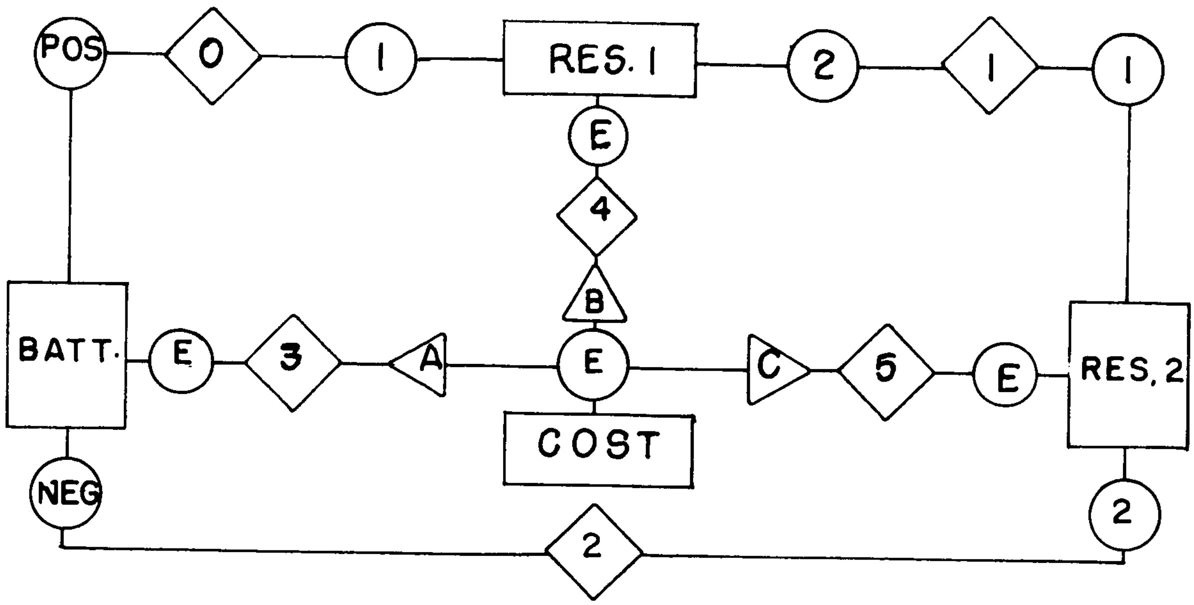


Figure C-5. Example System in Economics Simulator Format

```
 CONNECTIONS.
    BATT(1).TO.RES.1(2)
    RES.1(1).TO.RES.2(2)
    RES.2(1).TO.BATT(2)
    COST(E.A).TO.BATT(E)
    COST(E.B).TO.RES.1(E)
    COST(E.C).TO.RES.2(E)
 INPUT PARAMETERS.
    RIND(BATT(E)) = IP1
    RIND(RES.1(E)) = IP2
    RIND(RES.2(E)) = IP3
 DESIRED RESULTS.
    TCOST(COST(E)) = DR1
 NEW ELEMENT TAPE.
$            COMPILE MAD.PRINT OBJECT.PUNCH OBJECT
           R ECONOMICS SIMULATION
           R MUST READ THE RELIABILITY INDICES AS INPUT DATA
       START READ AND PRINT DATA
-            CONTINUE
           R FIRST PART OF CALCULATIONS GOES HERE
             T'O BEGIN
-       BACK CONTINUE
           R SECOND PART OF CALCULATIONS GOES HERE
             P'S DR1
             T'O START
-      BEGIN CONTINUE
             T'O BACK
+            E'M
 CALCULATIONS.
-            CONTINUE
+            CONTINUE
```

Figure C-6. Example System Source Program

4.2 ELEMENT DESCRIPTOR LIBRARY

The Element Descriptor Library (Figure C-7) contains Statement Collections for the economics simulation of the example system. These Statement Collections are, in fact, the postulated element cost functions.

```
ELEMENT DESCRIPTION.
    NAME OF ELEMENT = COST
    ATTACHMENT NAMES = E
    BROADSCOPE PARAMETERS = TCOST
STATEMENT COLLECTION.
    COST(E,A),THEN,TCOST(E)
             SUM=0.
             SUM=SUM+⊕COST(E,A)⊕
             ⊕TCOST(E)⊕=SUM
DESCRIPTION FINISHED.
ELEMENT DESCRIPTION.
    NAME OF ELEMENT = RES
    ATTACHMENT NAMES = 1,2,E
    BROADSCOPE PARAMETERS = VOLTS,TCOST
STATEMENT COLLECTION.
    RIND(I)=THEN,COST(E)
             ⊕COST(E)⊕=A/⊕RIND(E)⊕+B
DESCRIPTION FINISHED.
ELEMENT DESCRIPTION.
    NAME OF ELEMENT = BATT
    ATTACHMENT NAMES = 1,2,E
    BROADSCOPE PARAMETERS = VOLTS,TCOST
STATEMENT COLLECTION.
    RIND(E),THEN,COST(E)
             ⊕COST(E)⊕=C/⊕RIND(E)⊕+D
DESCRIPTION FINISHED.
```

Figure C-7. Example System Element Descriptions

5 CONCLUSIONS AND RECOMMENDATIONS

The core of the economics simulation problem lies in the generation of the element cost functions. The approach taken is to establish these functional relationships from available cost data through regression analyses. A major difficulty is the acquisition of a sufficient amount of valid cost data linking cost, reliability and performance.

Although the Westervelt Simulator may be legitimately applied to the system economics problem, it should not be considered to be the most efficient manner of solution. It is, therefore, recommended that further study be made in an effort to determine if other more desirable economics simulation procedures can be developed.

APPENDIX D

LAUNCH VEHICLE SYSTEM

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1 INTRODUCTION

1.1 OBJECTIVE

The objective of the launch vehicle study has been to provide a library of information and data that describes a hardware system. The library is required as an input to PRESTO*.

1.2 SCOPE

The scope of the study has been to supply information and data to a depth necessary to show application of PRESTO. The optimization of the vehicle operating as a system was based upon the analysis of performance, reliability, and economics. The study included the definition and description of a space launch vehicle, its mission objectives, and the preparation of functional characteristics of the elements that make up the system. Reliability estimates were prepared for each of the system elements. Costs were developed for the system, where cost was a function of specified performance and reliability.

As the material generated in this study was considered to be excessive for inclusion in the Final Report, only that matter which contributes to the basic understanding of the relationship between the system and the PRESTO concept is contained in this Appendix.

1.3 METHODOLOGY

The organization of material for systems analysis by means of the digital computer must be carefully disciplined and structured. The term "system" emphasizes that an over-all operational process is under consideration rather than a collection of pieces. Most systems defy easy comprehension by virture of inherent detail.

For large scale launch vehicles, a comprehensive description of the "over-all operational process" is quite complex. To avoid many burdensome and repetitious words, and to perceptually convey the method used to organize the material for the study, a "Construct of Vehicle Study" was prepared as shown in Figure D-1. This "construct" or concept provides insight into the heirarchy of the intersects of data and ideas which permit system analysis.

---

*Performance, Reliability, and Economic Simulation Techniques for Optimization

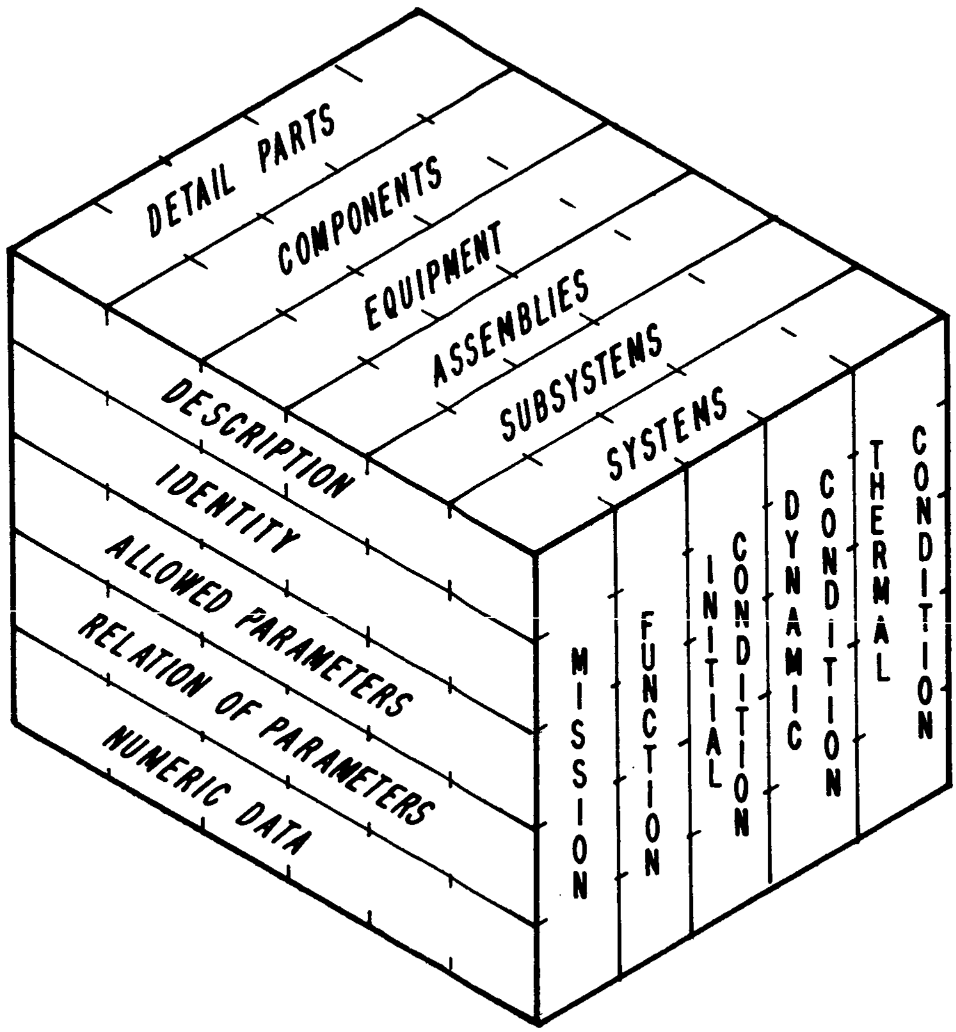


Figure D-1. Construct of Vehicle Study

By entering the cube of Figure D-1 at the word "Assemblies," and moving downward, each assembly will have a description, an identity, a set of allowed parameters, statements relating the parameters, and associated numeric data. In a like manner, progressing left-to-right in the cube, each assembly is related to the mission, each assembly has a function, and descriptions must exist for initial, dynamic, and terminal conditions. For any given item on the three planes, mobility is available to the other two planes to define intersects. Each intersect represents the conceptual "location" and type of data that is used in the study. The amount of data applicable to any given intersect is a function of the items and their relation to the overall mission.

This general approach to system description is deceptively simple. However, these items must be identified, since even the least detailed description usually proceeds by iteration. Each item is described roughly and then, subsequently, in increasing detail.

2 LAUNCH VEHICLE

In the demonstration of the optimization of a launch vehicle, it was deemed desirable to describe a system based upon an actual vehicle configuration. The selection of the vehicle upon which the study vehicle definition was to be fashioned was determined by the following factors:

- The vehicle system was to be representative of one that might be employed for many future missions.
- The vehicle system should provide a magnitude of complexity sufficient to demonstrate effectively the utility of the PRESTO concept.
- There should exist a considerable amount of readily available, unclassified information in the areas of system performance, reliability, and cost.

The stable of NASA vehicles was reviewed for candidate vehicles. The vehicle which most nearly met the desired criteria was the Atlas/Agena/Ranger configuration. Though a problem was posed by this choice of vehicle in the area of security classification, it was overcome by postulating data where security reasons so dictated. This resulted in an analysis which was somewhat removed from reality, but which in no way compromises the demonstration of the utility of the PRESTO methodology.

2.1 VEHICLE DEFINITION

The resultant Study Launch Vehicle System (SLVS) is a two and one-half stage, liquid-fueled missile, with a spacecraft payload section.

The SLVS first stage is identified as a BOGY vehicle, and is patterned after the Atlas LV-3 vehicle. The BOGY may be briefly described as a fixed thrust, liquid propellant, one and one-half stage vehicle employing:

- Two 150,000-lb thrust booster engines
- One 60,000-lb thrust sustainer engine

- Two 1,000-lb thrust vernier engines

- An on-board programmer for roll and pitch control

- A radio-inertial guidance system with an associated ground computer to generate steering, engine cut-off, and other discrete commands

The SLVS second stage is identified as a RAM vehicle; it has some characteristics similar to those of an Agena vehicle. The RAM is described as a fixed thrust, liquid propellant, one-stage vehicle employing:

- One 15,000-lb thrust engine

- Ullage rockets

- An adapter section with booster retro-rockets

- An "all-inertial" guidance system to sense and command vehicle performance

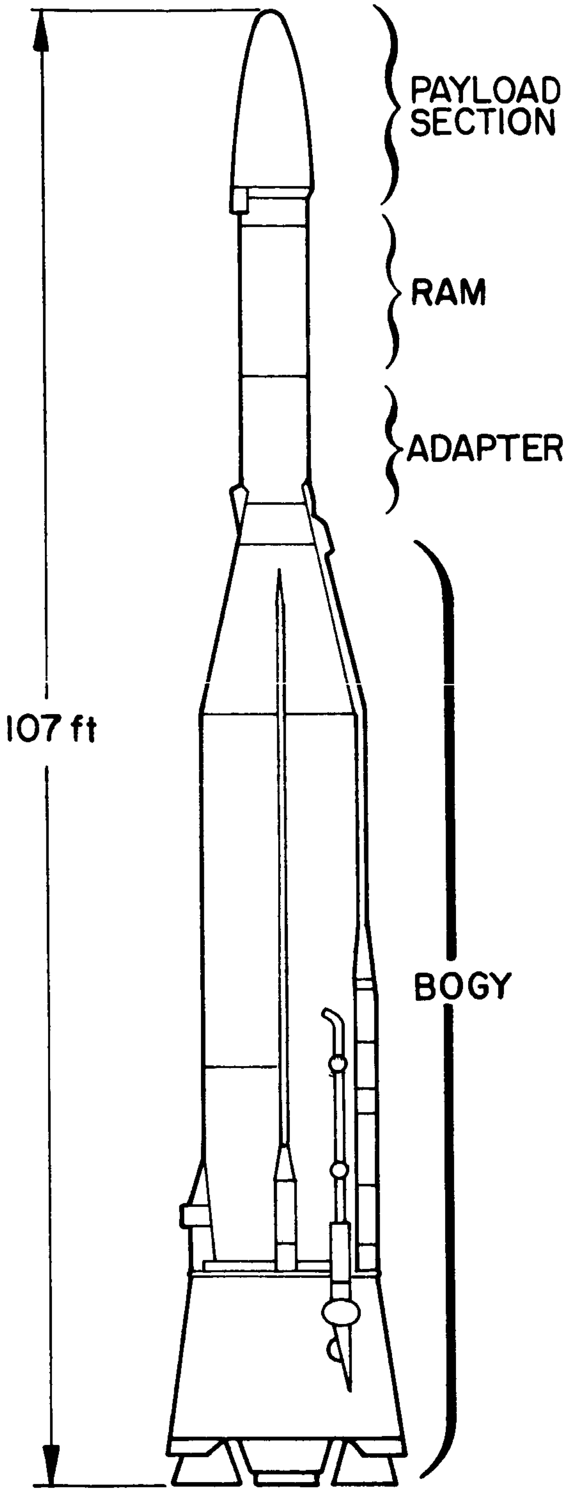
An outboard profile and general characteristics of the composite SLVS vehicle are shown in Figure D-2.

2.2 SLVS SUBSYSTEM DEFINITION

The SLVS is defined as containing seventeen prime subsystems. Of these seventeen subsystems, thirteen are BOGY subsystems and four and RAM subsytems, grouped as follows:

BOGY FIRST STAGE

- Booster Propulsion
- Booster Pneumatics
- Booster Hydraulics
- Booster Airframe
- Booster Separation
- Sustainer Propulsion
- Sustainer Hydraulics
- Sustainer Airframe
- Sustainer Pneumatics



STAGE

1ST stage-LOX/RP-1 (BOGY)
2ND stage-IRNFA/UDMH (RAM)

MISSION CAPABILITY

300 n. mi. orbit - 800 lbs
Lunar orbit - 80 lbs
Planitary probe - 50 lbs

USE

Lunar probes
Communications satellites
Scientific satellites

LAUNCH RATE CAPABILITY

10/yr/pad

Figure D-2. Profile of Composite Vehicle

- Propellant Utilization
- Electrical
- Guidance
- Flight Control

RAM SECOND STAGE

- Airframe
- Propulsion
- Electrical
- Guidance and Control

A diagram of the relationship of the launch vehicle subsystems appears in Figure D-3.

2.2.1 Subsystem Functional Description

- Booster Propulsion

  The Booster Propulsion Subsystem consists of the two booster engines and lower order equipment necessary to the operation of these engines. Each booster is a bi-propellant, liquid, fixed thrust, gimballed engine consisting of a tubular wall, fuel regeneratively cooled, hypergol-ignited thrust chamber supplied with missile tank-stored propellants by a hot gas generator-driven turbopump. Each engine is ground started by electrically firing a short duration solid propellant gas generator which accelerates the turbopump speed to bootstrap level and causes the fuel-actuated main propellant valves to open in accordance with a fuel pressure ladder sequence. Engine shutdown occurs at a guidance command to a solenoid-operated set of propellant valves. Fuel-rich shutdown is accomplished by having the oxidizer valve close slightly prior to the closing of the fuel valve. Booster No. 1 powers the booster hydraulic pump which supplies hydraulic power for gimballing the booster engines.

- Booster Pneumatics

  The main function of the Booster Pneumatics Subsystem is to provide the required pressurization of the propellant tanks during the boost phase of flight. The airborne pressurization subsystem comprises two subsystems, one for main propellant tank pressurization and one for pressurization and purges in the propulsion subsystem.

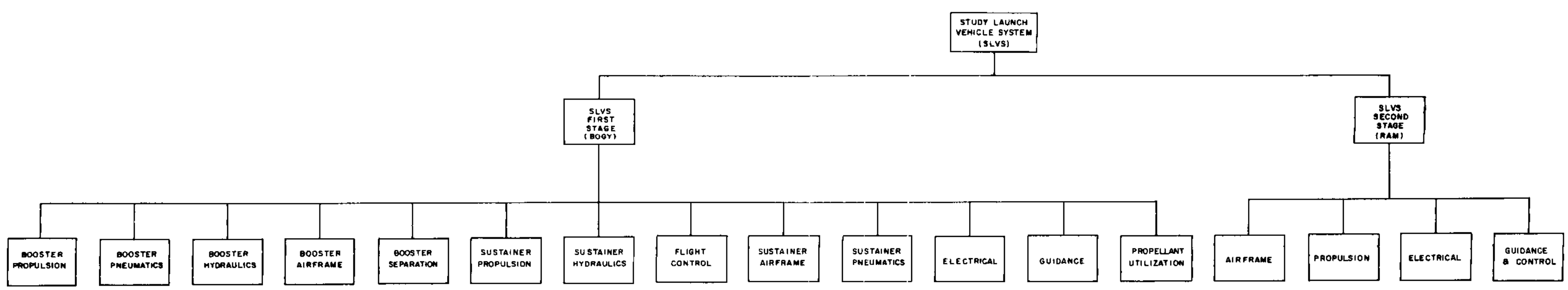


Figure D-3. SLVS Subsystem Diagram

Gas for tank pressurization during booster stage is supplied from six spherical containers located in the engine compartment and attached to the jettisonable thrust barrel. Connections between ground and missileborne equipment are made through mechanical disconnect valves which seal pressure in the missile after riseoff. A shutoff valve holds gas in the containers until the airborne subsystem is activated to allow maximum use of gas from ground facilities.

- Booster Hydraulics

This subsystem actuates the cylinders which control gimbals for pitch, roll, and yaw control of the vehicle during the boost phase of the flight. Hydraulic pumps driven from accessory drives of the engine turbopumps provide hydraulic pressure to the actuating cylinders. Also included in the Booster Hydraulics Subsystem are suitable manifolds, accumulators, tanks, and minor valves. Actuating commands are received from the Flight Control Subsystem and are translated from electrical to hydraulic commands by servo control valves mounted on the booster actuators.

- Booster Airframe

This structure consists of a thrust barrel, a shield, and a fairing installation. It forms a compartment to house the Propulsion Subsystem and its associated equipment.

- Booster Separation

This subsystem contains the components necessary to jettison the booster section at staging. It consists of separation fittings, a distribution manifold, a helium supply, supply lines, and separation explosive valves. At staging, an electrical signal is provided by the autopilot programmer which closes a circuit to the separation valves and causes a detonator to break a diaphram which will release high pressure helium gas to flow through the manifold to the separation fittings. This helium pressure then opens the latches by shearing pins. This allows the booster section to separate from the vehicle tank section by slipping aft along jettison tracks.

- Sustainer Propulsion

The Sustainer Propulsion Subsystem consists of the sustainer engine, the vernier engines, and supporting equipment. The sustainer engine is a gimbal-mounted unit installed on the thrust cone of the fuel tank. The assembly is similar to a booster engine except for the following principal deviations:

(1) During sustainer burn the sustainer engine turbopump supplies propellants to the vernier engines and gas generator as well as to the sustainer thrust chamber.

(2) The sustainer engine thrust chamber is smaller than those of the boosters.

Propellants for the vernier engines are supplied by the vernier solo tanks during starting and solo operation, and are supplied from the sustainer turbopump during mainstage operation.

- Sustainer Hydraulics

This subsystem actuates gimbal cylinders for the sustainer and vernier thrust chambers to provide pitch, roll, and yaw control of the vehicle during sustainer/vernier burn. Hydraulic pumps driven from accessory drives of the engine turbopump provide hydraulic pressure to the actuating cylinders. Also included in the Sustainer Hydraulics Subsystem are suitable manifolds, accumulators, tanks, and minor valves. Control orders are received from the Flight Control Subsystem and are translated from electrical to hydraulic commands by servo control valves mounted on each engine actuator.

- Sustainer Airframe

The propellant tank is the primary structure of the Sustainer Airframe Subsystem. The sustainer engine, with associated equipment, is gimbal-mounted to the sustainer thrust cone which is the aft end of the Sustainer Airframe. The vernier engine thrust chambers are gimbal-mounted on opposite sides of the Sustainer Airframe near the aft end of the structure.

- Sustainer Pneumatics

Fuel tank pressure is allowed to decay during sustainer stage and no gas supply is required. Vapor pressure adequately maintains tank pressure in the liquid oxygen tank after booster separation. Connections between the booster and sustainer subsystems are made through mechanical disconnect valves which seal gas in the sustainer section at staging.

Pressurization for the vernier engines during vernier solo stage is supplied by gas from a spherical container located in the engine compartment and attached to the nonjettisonable rails on the aft end of the fuel tank.

- Propellant Utilization

The Propellant Utilization Subsystem senses the mass of propellants remaining in the missile tanks, and adjusts the rate of fuel flow by repositioning the propellant valves to maintain a proper mass ratio of residual propellants.

- Electrical

The Electrical Subsystem provides electrical energy for all BOGY subsystems. The prime elements of the electrical subsystem are the 28 VDC battery, and the 115 VAC, 400 cycle inverter which is driven by the battery.

- Guidance

The Radio Inertial Guidance Subsystem determines the position and velocity of the vehicle and, on the basis of the trajectory objectives, commands the flight corrections necessary to fulfill these objectives. A ground-based monopulse tracking radar transmits composite bursts which contain identity information, interrogates an airborne position transponder, and commands discrete functions and attitude changes. A ground-based doppler transmitter interrogates an airborne rate transponder, the output of which is received by the ground receivers. Velocity is determined by the doppler shift of the ground-received signals. Position and velocity information flow to data extraction and processing circuitry in the ground computer, where commands are generated to accomplish trajectory objectives.

- Flight Control

The vehicle Flight Control Subsystem consists of a programmer, gyro package, servo amplifier/filter package, excitation transformer, and cabling subsystems. The integrated output of these subsystems steers and stabilizes the vehicle in performance of the mission objective.

The programmer is basically a timing device which contains the switches and controls that are necessary to properly sequence the vehicle's control functions. The programmer performs two primary functions:

(1) It implements the initial roll and pitch programs and initiates the staging command, and

(2) Times the issuance of programmed commands to the autopilot subsystem.

The displacement gyros generate error signals for each of the pitch, yaw, and roll channels. A given error signal is proportional to the integrated difference between the rate commands received from the programmer or Guidance Subsystem and the integrated body rate detected by the rate gyros.

Servo amplifiers control the position of the hydraulic actuators by comparing the commanded and actual engine positions. The filters serve to minimize the effect of undesirable body frequencies in the command signal.

The rate gyros are installed physically, in a separate package to minimize sensitivity to local bending modes. Body angle rates in pitch, yaw, and roll are sensed by the rate gyros and fed to the gyro signal amplifier to provide adequate rate damping of the command signals.

The excitation transformer provides a 400 cps reference voltage for excitation of the feedback transducers, and provides a voltage for vernier bias signals after staging.

Cabling consists of the inter-subsystem electrical and signal cabling and associated connectors and plugs.

- RAM Airframe

  This subsystem acts as the support structure and container for all the second stage subsystems. Items included are the equipment mountings, tankage, fairings, shields, and the upper-stage adapter section.

- RAM Propulsion

  The RAM propulsion subsystem consists of a single main liquid-fueled thrust chamber, and the associated propellant system. The engine is gimbal mounted for thrust vector control. A propellant control system meters the proper flow of propellants; proper flow is enhanced by a propellant pressurization system. Second-stage ullage rockets and the booster retro-rockets are considered part of the Propulsion Subsystem.

- RAM Electrical

  All second stage electrical power and power distribution items are included in the Electrical Subsystem. Included are primary and secondary batteries, power supplies, power control circuits, junction boxes, and cabling.

- RAM Guidance and Control

  The RAM Guidance and Control (G and C) subsystem provides the attitude control system for the second stage stabilization and steering. Active attitude control of the second stage is inhibited until after separation from the BOGY. During coast periods, stabilization is maintained by a pneumatic attitude control system; during powered flight of the second stage, stabilization and steering is maintained by a hydraulic attitude control system. Commands for attitude control and engine control are generated within program sequence timers, an on-board guidance computer, a horizon sensor package, and "flight-control electronics."

2.3 MISSION

The primary mission objective for the SLVS is to inject an 80-pound payload into the proper lunar intercept trajectory.

2.3.1 BOGY Booster Objective

The primary mission of the BOGY booster is to place the RAM/spacecraft at the proper attitude and at a predetermined position and velocity in space as defined by the appropriate guidance equations.

2.3.2 RAM Objective

The RAM second stage mission objective is to inject the spacecraft into the prescribed lunar transfer orbit, which is determined by vehicle velocity and position at RAM thrust termination.

2.4 FLIGHT PLAN

After completion of the countdown, the engine ignition sequence is initiated. The BOGY booster, sustainer, and vernier engines are ignited. When mainstage engine thrust buildup is complete, the launcher release sequence is automatically initiated, and the vehicle rises through a short controlled-release period. As the vehicle lifts from the pad, the umbilical connections are separated by pneumatic and electrical forces and lanyard pull-away. After release, and continuing through BOGY retro-rocket ignition, all BOGY sequences are controlled by the flight programmer. Discrete commands are provided by a radio guidance system to control certain primary functions; the steering and discrete commands flow to the flight programmer, and thence to the points of control.

The composite vehicle will lift off with azimuth of 105 degrees true and ascends vertically for the first 15 seconds of flight. After two seconds of vertical motion, a programmed roll maneuver aligns the vehicle with the required trajectory azimuth, which is established by the date and time of launch. At the end of 15 seconds, the roll program is terminated and a programmed pitch maneuver is initiated and will continue until BOGY staging. BOGY staging is actuated by the flight programmer at approximately 126 seconds; the command is executed in response to a radio guidance discrete at approximately 136 seconds. The autopilot programmer will then command booster engine cutoff (BECO) and booster thrust section jettison.

The vehicle will continue under BOGY-powered flight through sustainer engine cutoff and up to BOGY vernier engine cutoff. Sustainer engine cutoff is actuated by the flight programmer at booster engine cutoff plus 60 seconds, and the radio guidance discrete command will be received at approximately 264 seconds when the attitude, velocity, and position criteria programmed in the ground guidance computer are satisfied. Between sustainer engine cutoff (SECO) and vernier engine cutoff (VECO) (a period of approximately 16 seconds), stability and attitude of the vehicle will be controlled by the thrust from the vernier engines.

Following sustainer engine cutoff, but prior to VECO, at approximately 270 seconds, the RAM program timer will be started by a radio guidance discrete.

Vernier engine cutoff will occur at approximately 280 seconds in response to a radio guidance discrete. One second after VECO, the RAM nose shroud pin puller squibs will be fired by a radio guidance discrete and the shroud will be ejected. The shroud will be allowed to travel for two seconds before BOGY/RAM separation, to assure clearance from the RAM/spacecraft combination.

BOGY/RAM separation will occur at approximately 283 seconds when RAM midbody pin pullers are fired by a BOGY guidance discrete. At the same time, BOGY retro-rockets will be fired and the BOGY will be retarded, permitting the RAM/Spacecraft to draw free of the adapter.

The firing of the BOGY retro-rockets terminates the BOGY participation in the overall mission. The remnant BOGY vehicle will be in free-fall from this time on, and will return to the earth's surface in the sustainer tankage impact area.

Mission requirements of the BOGY as the primary booster for the upper stages are incorporated in a set of inertial velocity parameters existing at BOGY vernier cutoff. Fulfillment of the flight requirements at the termination of BOGY-powered flight allows the RAM/ spacecraft vehicle to coast to the proper space position with the proper velocity and flight path angles for the firing of the RAM engine which places the vehicle into the desired trans-lunar trajectory.

Following separation, the RAM/spacecraft will coast for 40 seconds. Attitude control during this time is available through a set of fixed gas jets.

At 322 seconds, RAM ullage rockets are ignited to assure that the propellants are properly oriented to avoid cavitation of pumps at initiation of main engine burn. The main engine burn begins at 332 seconds. During powered flight, attitude control is accomplished by hydraulically actuated gimballing of the thrust chamber. In this mission, the powered flight of RAM terminates with engine cutoff at 449 seconds, as established by the on-board guidance computer.

3 LAUNCH VEHICLE PERFORMANCE*

Application of the PRESTO methodology dictates the need for mathematical models of the performance characteristics of the SLVS in terms of the performance characteristics of the various components which make up the system. The generation of such models has been attempted through use of the Performance Simulator program as developed at the University of Michigan under Dr. F. H. Westervelt. The information that is required by the Westervelt Simulator consists of the following:

a. Source Program
b. Element Descriptor Library.

The Source Program describes the particular problem under consideration to the Westervelt Simulator. The Element Descriptions describe the physical laws or relationships for each element in the system to the Simulator. The rules for communicating with the Simulator and a complete discussion of the logic used are delineated in Part A of Volume III of this report.

In the SLVS study, two levels for performance simulation were selected:

a. Simulation at the subsystem level, and
b. Simulation at the system level.

The initial efforts were to have been directed primarily toward the generation of the Source Program and Element Descriptions required at the subsystem level. Upon completion of tasks at the subsystem level, the subsystem information was to be incorporated in the simulation of performance of the SLVS as a complete system.

3.1 SUBSYSTEM STUDY

The BOGY Pneumatics Subsystem was selected as the first subsystem to be analyzed. The development of Simulator information (Source Program and Element Descriptions) for the BOGY Pneumatics Subsystem

*It should be noted that only the descriptions of systems pertaining to the SLVS are contained in this Appendix. Several other systems were reduced to the proper Simulator format and were presented for simulation to assist in the location of the problem areas of the Westervelt Simulator.

was divided into two phases. In the first phase, a much simplified Pneumatics Subsystem was to be developed and simulated. The second phase of the Pneumatics Subsystem study was to include a complete BOGY Pneumatics Subsystem which is comprised of both the Booster Pneumatics and Sustainer Pneumatics Subsystems discussed in Section 2 of this Appendix.

3.1.1 Simplified Pneumatics Subsystem

Although the Simplified Pneumatics Subsystem has little direct correspondence to the BOGY subsystem, it is representative of a propellant tank pressurization system of a liquid-fueled missile. A diagram of the Simplified Pneumatics Subsystem in Westervelt Simulator format appears in Figure D-4. This diagram defines the subsystem as a composition of the elements and their interconnections. These elements are as follows:

- AIRFRM — Airframe
- OTANK — Oxidizer Tank
- FTANK — Fuel Tank
- FREG — Fuel Tank Pressure Regulator
- OOUT — Oxidizer Sink
- FOUT — Fuel Sink
- OXREG — Oxidizer Tank Pressure Regulator
- OFREG — Oxidizer Tank - Fuel Tank Differential Pressure Regulator
- HEX — Heat Exchanger
- HELS — Helium Source
- BTEMP — Booster Temperature into Heat Exchanger

The Source Program and Element Descriptions for the subsystem were prepared in accordance with the rules specified in Part A of Volume III and appear in Figure D-5 and Figure D-6, respectively.

The Simplified Pneumatics Subsystem was presented for performance simulation with the Westervelt Simulator. However, for reasons discussed in Section 5.2.2 of Volume I, the simulation was not successful.

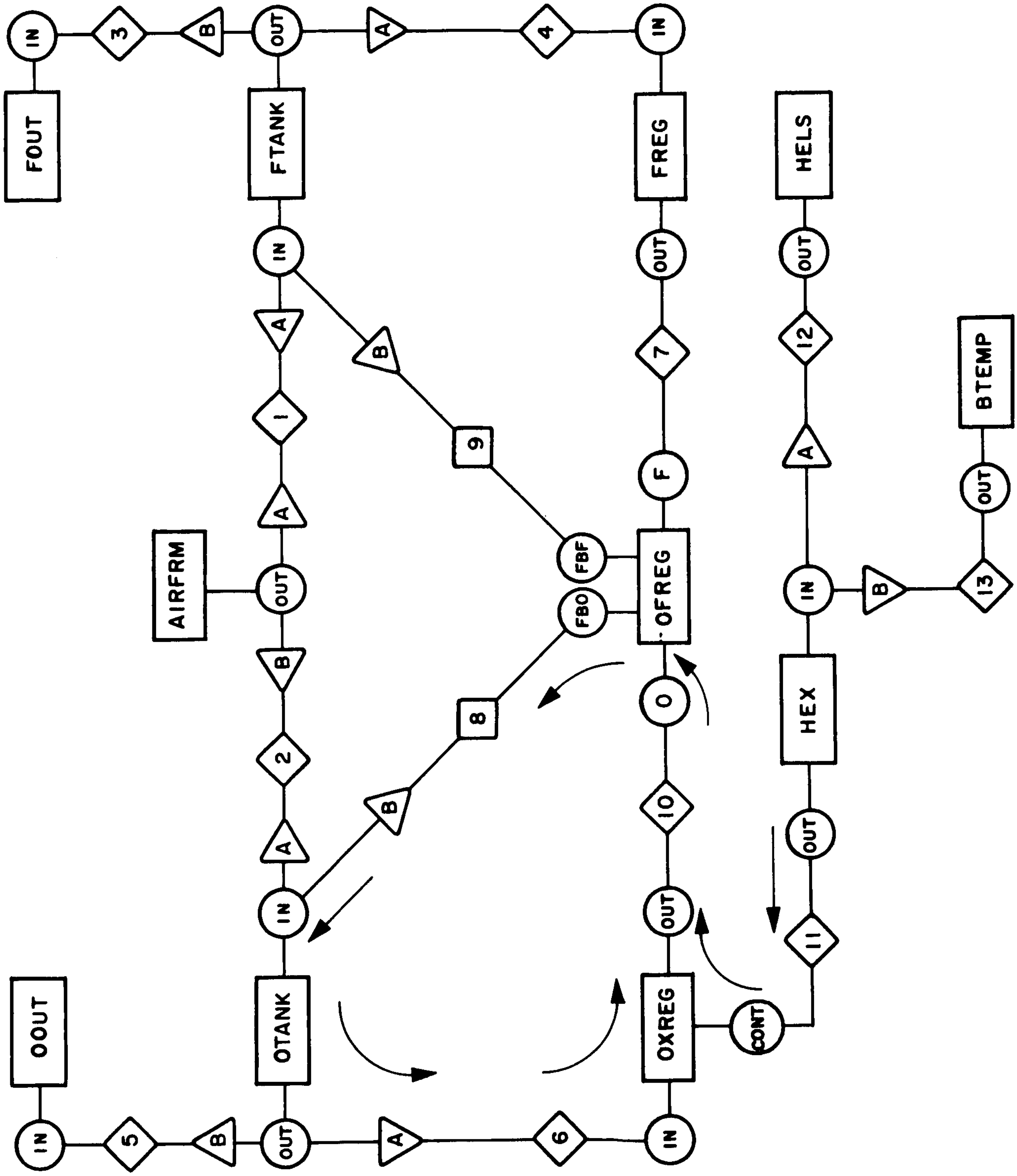


Figure D-4. Simplified Pneumatics Subsystem

```
IDENT=$PNEMAT$   *
CONNECTIONS.
    AIRFRM(OUT,A),TO,FTANK(IN,A)
    AIRFRM(OUT,B),TO,OTANK(IN,A)
    FTANK(OUT,B),TO,FOUT(IN)
    FTANK(OUT,A),TO,FREG(IN)
    OTANK(OUT,B),TO,OOUT(IN)
    OTANK(OUT,A),TO,OXREG(IN)
    FREG(OUT),TO,OFREG(F)
    OFREG(FBO),TO,OTANK(IN,B)
    OFREG(FBF),TO,FTANK(IN,B)
    OXREG(OUT),TO,OFREG(O)
    HEX(OUT),TO,OXREG(CONT)
    HEX(IN,A),TO,HELS(OUT)
    HEX(IN,B),TO,BTEMP(OUT)


INPUT PARAMETERS.
    VBL(AIRFRM(OUT))
    VBH(AIRFRM(OUT))
    TBO(AIRFRM(OUT))
    DOA(OTANK(IN))
    DOF(FTANK(IN))
    DFP(FREG(IN))
    DOP(OXREG(IN))
    DOFP(OFREG(O))
    IFTP(FTANK(IN))
    IOTP(OTANK(IN))


DESIRED RESULTS.
    PRES(OFREG(O))
    PRES(OFREG(F))


EQUIVALENCE.
    DR1=PRES(OFREG(O))
    DR2=PRES(OFREG(F))


NEW ELEMENT TAPE.
$              COMPILE MAD,PRINT OBJECT,PUNCH OBJECT
               STATEMENT LABEL DONE1,DONE2,STAGE
               BOOLEAN FRT,FIRST
               V'S  FIRST=1B
               V'S  FRT=1B
 START         CONTINUE
              R MUST READ TMAX,ROP,RFP,ROFR AS INPUT DATA
               READ AND PRINT DATA
-              DT=0
              RFIRST PART OF CALCULATIONS GOES HERE
               T'O BEGIN
-        BACK CONTINUE
              RSECOND PART OF CALCULATIONS GOES HERE
              P'S TIME,DR1,DR2
              T'O START
-     BEGIN CONTINUE
              T'O BACK
              E'M
 CALCULATIONS.
-             THROUGH MAINLP,FOR TIME=0.,DT,TIME.GE.TMAX
              DT=TIME.(TIME)
              TIME=TIME+DT
+   MAINLP CONTINUE
```

Figure D-5. Simplified Pneumatics Subsystem Source Program

```
ELEMENT DESCRIPTION.
NAME OF ELEMENT = AIRFRM
ATTACHMENT NAMES = OUT
BROAD SCOPE PARAMETERS = PRES.VIBR.TEMP.VBL.VBH.TBO
STATEMENT COLLECTION.
   VBL.VBH.TBO(OUT).THEN.VIBR(OUT)
            W'R TIME.LE.@TBO(OUT)@
            @VIBR(OUT)@=@VBH(OUT)@
            O'E
            @VIBR(OUT)@=@VBL(OUT)@
            E'L
STATEMENT COLLECTION.
   COMPUTE.X2D.Y2D.Z2D(OUT).TEMP(OUT)
            AFM.(@X2D(OUT)@.@Y2D(OUT)@.@Z2D(OUT)@.@TEMP(OUT)@)
DESCRIPTION FINISHED.

ELEMENT DESCRIPTION.
NAME OF ELEMENT = OTANK
ATTACHMENT NAMES = IN.OUT
BROAD SCOPE PARAMETERS = PRES.VIBR.TEMP.IOTP.DOA
STATEMENT COLLECTION.
   VOL.DOA.TEMP.IOTP(IN).THEN.PRES(IN)
            W'R FIRST
            FIRST=OB
            @PRES(IN)@=@IOTP(IN)@
            O'E
            @PRES (IN)@=@DOA(IN)@*1545.*@TEMP(IN)@
STATEMENT COLLECTION.
   TEMP(IN).PRES(IN).PRES(OUT).THEN.TEMP(OUT)
            @TEMP(OUT)@=@PRES(OUT)@*@TEMP(IN)@/@PRES(IN)@
STATEMENT COLLECTION.
   UVOL(OUT).VOL(IN).THEN.VOL(OUT)
            @VOL(OUT)@=@VOL(IN)@-@UVOL(OUT)@
STATEMENT COLLECTION.
   VOL(OUT).TEMP(OUT).DOA(IN).THEN. PRES(OUT)
            TM=@VOL(OUT)@*@DOA(IN)@
            @ PRES(OUT)@=TM*1545.*@TEMP(OUT)@/@VOL(OUT)@
DESCRIPTION FINISHED.

ELEMENT DESCRIPTION.
NAME OF ELEMENT = FTANK
ATTACHMENT NAMES = IN.OUT
BROAD SCOPE PARAMETERS = PRES.VIBR.TEMP.IFTP.DOF
STATEMENT COLLECTION.
   VOL.DOF(IN).TEMP(IN).IFTP(IN).THEN.PRES(IN)
            W'R FRT
            FRT=OB
            @PRES(ON)@=@IFTP(IN)@
            O'E
            @ PRES(IN)@=@DOF(IN)@*1545.*@TEMP(IN)@
            E'L
STATEMENT COLLECTION.
   UVOL(OUT).VOL(IN).THEN.VOL(OUT)
            @VOL(OUT)@=@VOL(IN)@-@UVOL(OUT)@

STATEMENT COLLECTION.
   VOL(OUT).TEMP(OUT).DOF(IN).THEN. PRES(OUT)
            TF=@VOL(OUT)@*@DOF(IN)@
            @ PRES(OUT)@=TF*1545.*@TEMP(OUT)@/@VOL(OUT)@
DESCRIPTION FINISHED.
```

Figure D-6. Simplified Pneumatics Subsystem Element Descriptions (1 of 3)

```
ELEMENT DESCRIPTION.
NAME OF ELEMENT = FREG
ATTACHMENT NAMES = IN.OUT
BROAD SCOPE PARAMETERS = PRES.VIBR.DFP.TEMP
STATEMENT COLLECTION.
   PRES(IN).DFP(IN).THEN.PRES(OUT)
           TR=@PRES(IN)@-@DFP(IN)@
           W'R  TR .LE. RFP
           @ PRES(OUT)@=@DFP(IN)@
           O'E
           @PRES(OUT)@=@PRES(IN)@-(TR-RFP)
           E'L
DESCRIPTION FINISHED.

ELEMENT DESCRIPTION.
NAME OF ELEMENT = OOUT
ATTACHMENT NAMES = IN
BROAD SCOPE PARAMETERS = PRES.VIBR
STATEMENT COLLECTION.
   COMPUTE.UVOL(IN)
           OXG.(@UVOL(IN)@)
DESCRIPTION FINISHED.

ELEMENT DESCRIPTION.
NAME OF ELEMENT = FOUT
ATTACHMENT NAMES = IN
BROAD SCOPE PARAMETERS = PRES.VIBR
STATEMENT COLLECION.
   COMPUTE.UVOL(IN)
           FEUL.(@UVOL(IN)@)
DESCRIPTION FINISHED.

ELEMENT DESCRIPTION.
NAME OF ELEMENT = OXREG
ATTACHMENT NAMES = IN.OUT.CONT
BROAD SCOPE PARAMETERS = PRES.VIBR.DOP.TEMP
STATEMENT COLLECTION.
   PRES(IN).PRES(CONT).DOP(IN).THEN.PRES(OUT)
           TS=@PRES(IN)@+@PRES(CONT)@-@DOP(IN)@
           W'R  TS.LE. ROP
           @ PRES(OUT)@=@DOP(IN)@
           O'E
           @ PRES(OUT)@=@ PRES(OUT)@-(TS-ROP)
           E'L
DESCRIPTION FINISHED.

ELEMENT DESCRIPTION.
NAME OF ELEMENT = OFREG
ATTACHMENT NAMES = O.F.FBO.FBF
BROAD SCOPE PARAMETERS = PRES.VIBR.TEMP.DOFP
STATEMENT COLLECTION.
   DOFP(O).PRES(O).PRES(F).THEN.PRES(FBO).PRES(FBF)
           W'R TIME.GE. TMS T'O@**DONE1@
           TT=@ PRES(O)@-@ PRES(F)@
           W'R  TT.LE. ROFR
           @PRES(FBO)@=@DOFP(O)@
           O'E
           @PRES(FBO)@=@ PRES(O)@-(TT-ROFR)
           @PRES(FBF)@=@ PRES(F)@
           E'L
           T'O@**DONE2@
```

Figure D-6. Simplified Pneumatics Subsystem Element Descriptions (2 of 3)

```
⊕**DONE1⊕   ⊕PRES(FBO)⊕=⊕PRES(O)⊕
            ⊕PRES(FBF)⊕=⊕PRES(F)⊕
⊕**DONE2⊕   CONTINUE
 STATEMENT COLLECTION.
    DOFP(O).PRES(O).PRES(F).THEN.PRES(FBO).PRES(FBF)
            W'R TIME .GE. TMS T'O ⊕**DONE1⊕
            TT=⊕ PRES(O)⊕-⊕ PRES(F)⊕
            W'R TT.LE.ROFR
            ⊕PRES(FBF)⊕=⊕DOFP(O)⊕
            O'E
            ⊕PRES(FBF)⊕=⊕ PRES(F)⊕-(TT-ROFR)
            ⊕PRES(FBO)⊕=⊕ PRES(O)⊕
            E'L
            T'O⊕**DONE2⊕
⊕**DONE1⊕     ⊕PRES(FBO)⊕=⊕ PRES(O)⊕
            ⊕PRES(FBF)⊕=⊕ PRES(F)⊕
⊕**DONE2⊕     CONTINUE
 DESCRIPTION FINISHED.

ELEMENT DISCRIPTION.
 NAME ELEMENT = HEX
 ATTACHMENT NAMES = IN.OUT
 BROAD SCOPE PARAMETER = PRES.VIBR.TEMP
 STATEMENT COLLECTION.
    PRES(IN).TEMP(IN).THEN.PRES(OUT)
            W'R TIME .GE. TMS T'O  ⊕**STAGE⊕
            ⊕PRES(OUT)⊕=F(⊕TEMP(IN)⊕.⊕PRES(IN)⊕)
⊕**STAGE⊕     CONTINUE
 DESCRIPTION FINISHED.

 ELEMENT DESCRIPTION.
 NAME OF ELEMENT = HELS
 ATTACHMENT NAMES = OUT
 BROAD SCOPE PARAMETERS = PRES.VIBR.TEMP
 STATEMENT COLLECTION
    COMPUTE.PRES(OUT)
            HELP.(⊕PRES(OUT)⊕)
 DESCRIPTION FINISHED.

 ELEMENT DESCRIPTION.
 NAME ELEMENT = BTEMP
 ATTACHMENT NAMES = OUT
 BROAD SCOPE PARAMETERS = PRES.VIBR.TEMP
 STATEMENT COLLECTION.
    COMPUTE.TEMP(OUT)
            BTMP.(⊕TEMP(OUT)⊕)
 DESCRIPTION FINISHED.
            NEXT SET OF DATA.
```

Figure D-6. Simplified Pneumatics Subsystem Element Descriptions (3 of 3)

3.1.2 Tanks Pressurization Subsystem

During the period through which attempts were being made to obtain successful performance simulation of the Simplified Pneumatics Subsystem with the Westervelt Simulator, considerable effort was expended towards defining the Tanks Pressurization Subsystem. The Tanks Pressurization Subsystem has high correspondence to the BOGY Pneumatics Subsystem, and was defined as shown in Figure D-7. The elements appearing in the figure are identified as follows:

- AFME — Airframe
- FTNK — Fuel Tank
- OTNK — Oxidizer Tank
- REG, OT — Regulator, Oxidizer Tank
- REG, FT — Regulator, Fuel Tank
- PV — Poppet Valve
- RV — Relief Valve
- CV — Check Valve
- SV — Shut-off Valve
- GB — Gas Bottle
- HXB — Heat Exchanger, Booster
- HXA — Heat Exchanger, Ambient
- ATMOS — Atmosphere
- TRADP — Differential Pressure Transducer
- SWDP — Differential Pressure Switch

The element description for "valves" posed a significant problem. Because of the search logic of the Simulator, the element descriptions require the provision to permit redirection of the search for certain classes of problems. If flow through a valve ceases, the values of certain variables are "reflected" upstream and downstream to attached elements. But the

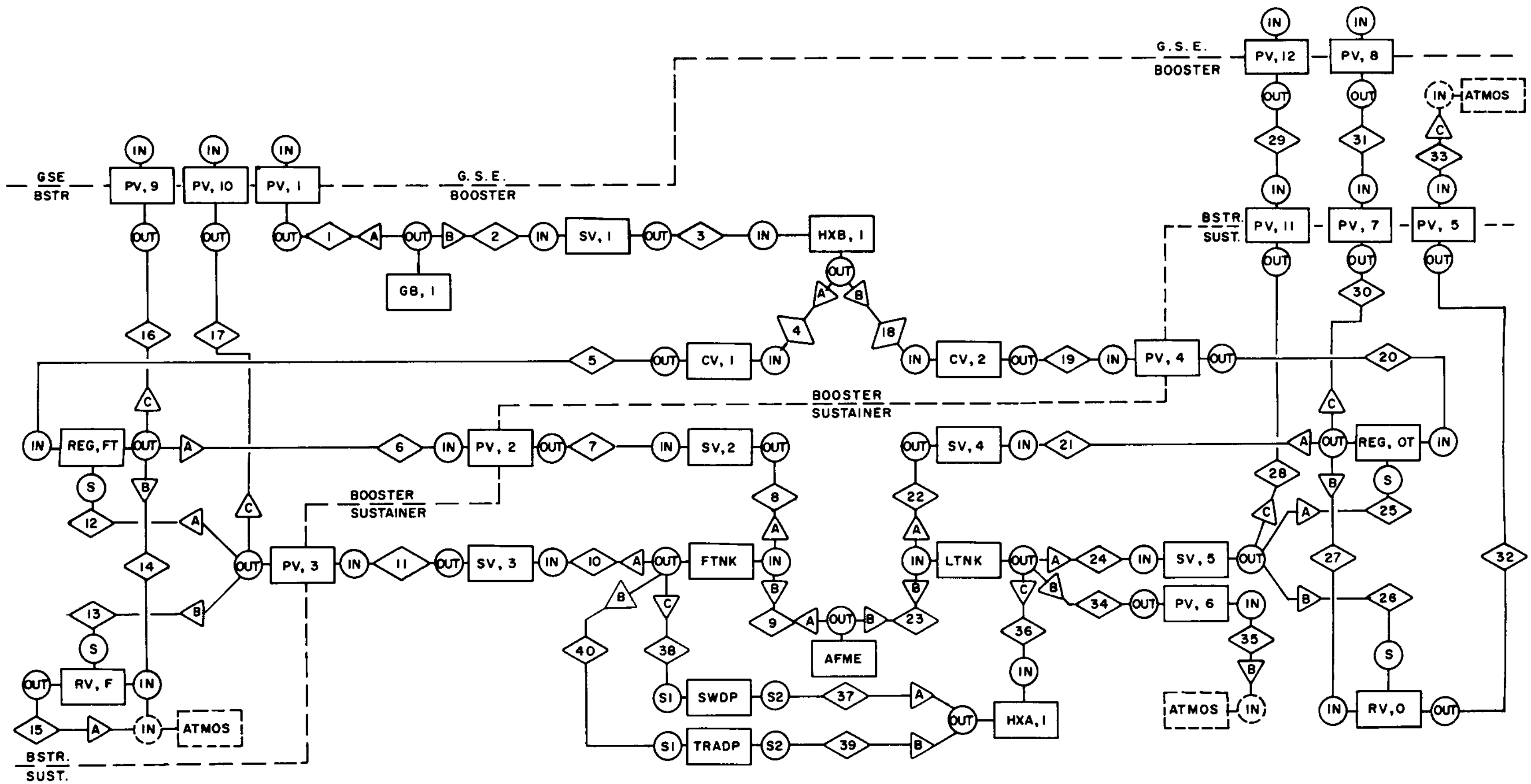


Figure D-7. BOGY Tanks Pressurization Subsystem values of variables at the attached elements may be dependent upon the mode of the valve. This situation revealed a weakness in the concept that an element could be adequately described as an entity, independent of its application in a system. Attempts were made to write acceptable descriptions for a simple electrical switch, without encouraging results. Before the valve/switch impasse could be resolved, it had become apparent that the Westervelt Simulator in its current state could not solve the earlier-submitted Simplified Pneumatics Subsystem problem. Efforts directed toward the development of the element descriptions for the Tanks Pressurization Subsystem were curtailed. However, at that time, it was hoped that the difficulties within the Simulator might be corrected and sufficient improvements incorporated such that a simulation of the BOGY at the system level might be achieved.

3.2 SYSTEM-LEVEL STUDY

At the time the subsystem level of study was abandoned, it was relatively late in the contract period. Investigation and application of alternate methods of performance simulation were deemed to pose too great a compromise of the remaining tasks of the PRESTO study. The goal of system optimization retained the requirement for the simulation of performance, as well as for reliability and economics. With an established dependency upon the Westervelt simulator, it was decided to attempt performance simulation at the "system" level.

The system investigated was the BOGY vehicle in the sustainer/vernier configuration. The analysis was to begin at lift-off plus 140-seconds, after the booster package had been jettisoned. This greatly simplified the Source Program and the Element Descriptions.

Since the individual subsystems could not be modeled adequately, the BOGY vehicle was defined in a slightly different manner. The elements allowed in the description were oriented toward system functions rather than as discrete subsystems.

The BOGY vehicle in the sustainer/vernier configuration was defined as shown in Figure D-8. The block diagram defines the system in terms of the functional elements and their interconnections.

The elements of the BOGY vehicle and their definitions follow:

- Airframe (AFME) — The airframe is the container for all the other elements, and acts as the structural support. All internal and external forces are transmitted through the airframe.

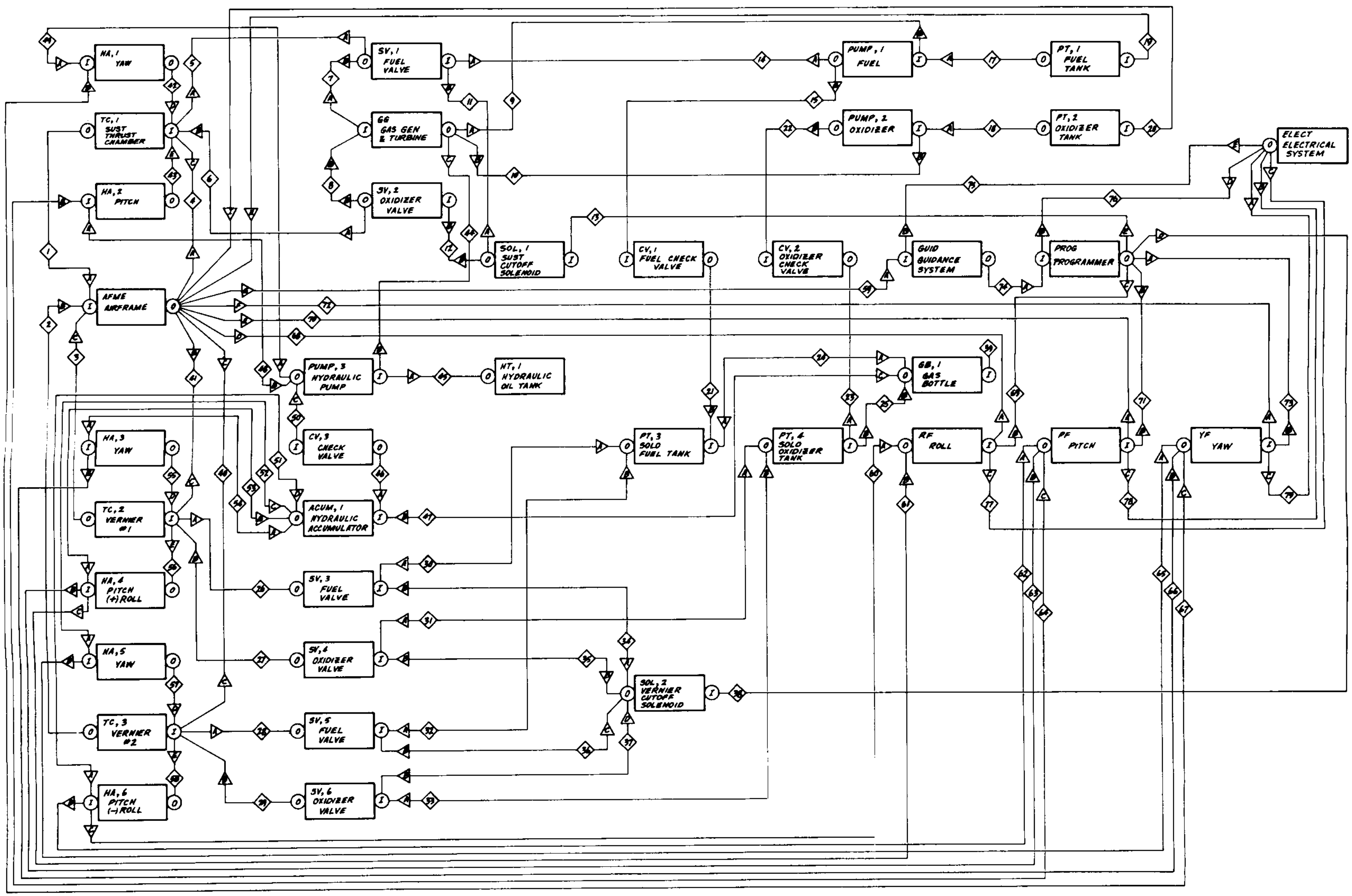


Figure D-8. BOGY Sustainer Vehicle Block Diagram

- Thrust Chamber (TC) — The thrust chamber is the assembly in which mixing of liquid propellants takes place to form hot gases which are then ejected through a nozzle at high velocity to give momentum to the system.

- Gas Generator (GG) — The gas generator is a combustion chamber used to provide hot gases for a turbine to drive the propellant pumps. In the present description, the gas generator and turbine characteristics are combined in one element description.

- Solenoid Valve (SV) — The solenoid valve is used to control the flow of a liquid or gas. The valve is mechanically operated by a coupling to a solenoid.

- Pump (PUMP) — Pumps are used to provide the proper flow rates of propellants. The pumps are driven by the gas generator/turbine assembly.

- Propellant Tanks (PT) — Propellant tanks are used to contain the propellants. In the instance of the BOGY these tanks are part of the airframe itself, but are defined as separate elements for the library. The tanks are pressurized to maintain the structural integrity of the airframe, and to prevent cavitation of the propellant pumps.

- Check Valve (CV) — Check valves permit the flow of a fluid in one direction only. A negative pressure differential must exist in the direction of the desired flow.

- Solenoid (SOL) — The solenoid is an electrically-operated device to produce a mechanical force through magnetic coupling.

- Gas Bottle (GB) — The gas bottle is a container to act as a source of pressurized gas. In the system as defined herein, the gas is used for pressurization of the propellant tanks.

- Hydraulic Actuator (HA) — The hydraulic actuators serve to position the gimballed thrust chambers to control the thrust vectors.

- Hydraulic Accumulator (ACUM) — The hydraulic accumulator serves to reduce the effects of surges through the hydraulic distribution lines.

- Hydraulic Oil Tank (HT) — The hydraulic oil tank is the source for hydraulic fluid.

- Guidance System (GUID) — The guidance system furnishes pitch and yaw steering commands during sustainer/vernier flight, and furnishes discrete commands for configuration changes.

- Flight Programmer (PROG) — The flight programmer furnishes roll and pitch programs, receives the guidance steering and discrete commands and routes them to the proper functional areas, and generates back-up commands for certain functions.

- Roll Function (RF)* — Roll function generates the signals to control the thrust vectors in the roll plane.

- Pitch Function (PF) * — Pitch function generates the signals to control the thrust vectors in the pitch plane.

- Yaw Function (YF) * — Yaw function generates the signals to control the thrust vectors in the yaw plane.

---

* Roll, pitch, and yaw functions actually are composed of displacement and rate gyros, and servo-amplifiers/filters for the three functions. Guidance steering in yaw and pitch is accomplished through torquing currents in the yaw and pitch displacement gyros of the autopilot.

- <u>Electrical System (ELECT)</u> — The electrical system furnishes AC and DC power to all user systems.

Soon after the start of the writing of the Element Descriptions for the BOGY Sustainer Vehicle, it became apparent that the number of parameters required to describe the elements of the system would far exceed the limit prescribed by the Westervelt Simulator. Consequently, the decision was made to reduce the BOGY Sustainer Vehicle into a Modified Sustainer Vehicle.

3.2.1 Modified Sustainer Vehicle

The Modified Sustainer Vehicle is comprised of elements which represent essentially four subsystems;

a. Sustainer Propulsion,
b. Sustainer Airframe,
c. Sustainer Pneumatics, and
d. ~~Propellant Utilization (see Section 2.0 of this Appendix).~~

A description of the Modified Sustainer Vehicle appears in Figure D-9. The definitions of the system elements are identical to those appearing in Section 3.2. The Source Program and Element Descriptions which were prepared for the Modified Sustainer Vehicle appear in Figure D-10 and Figure D-11, respectively. A list of parameters which appear in the element descriptions, together with the definitions of these parameters, are given in Figure D-12.

Though considerable effort was expended on the Westervelt Simulator at the University of Michigan, the problems within the Simulator, some of which are discussed in Sections 4.1 and 5.2.2.1.1 of this Final Report, remained unsolved. Consequently, the performance simulation of the Modified Sustainer Vehicle resulted in failure.

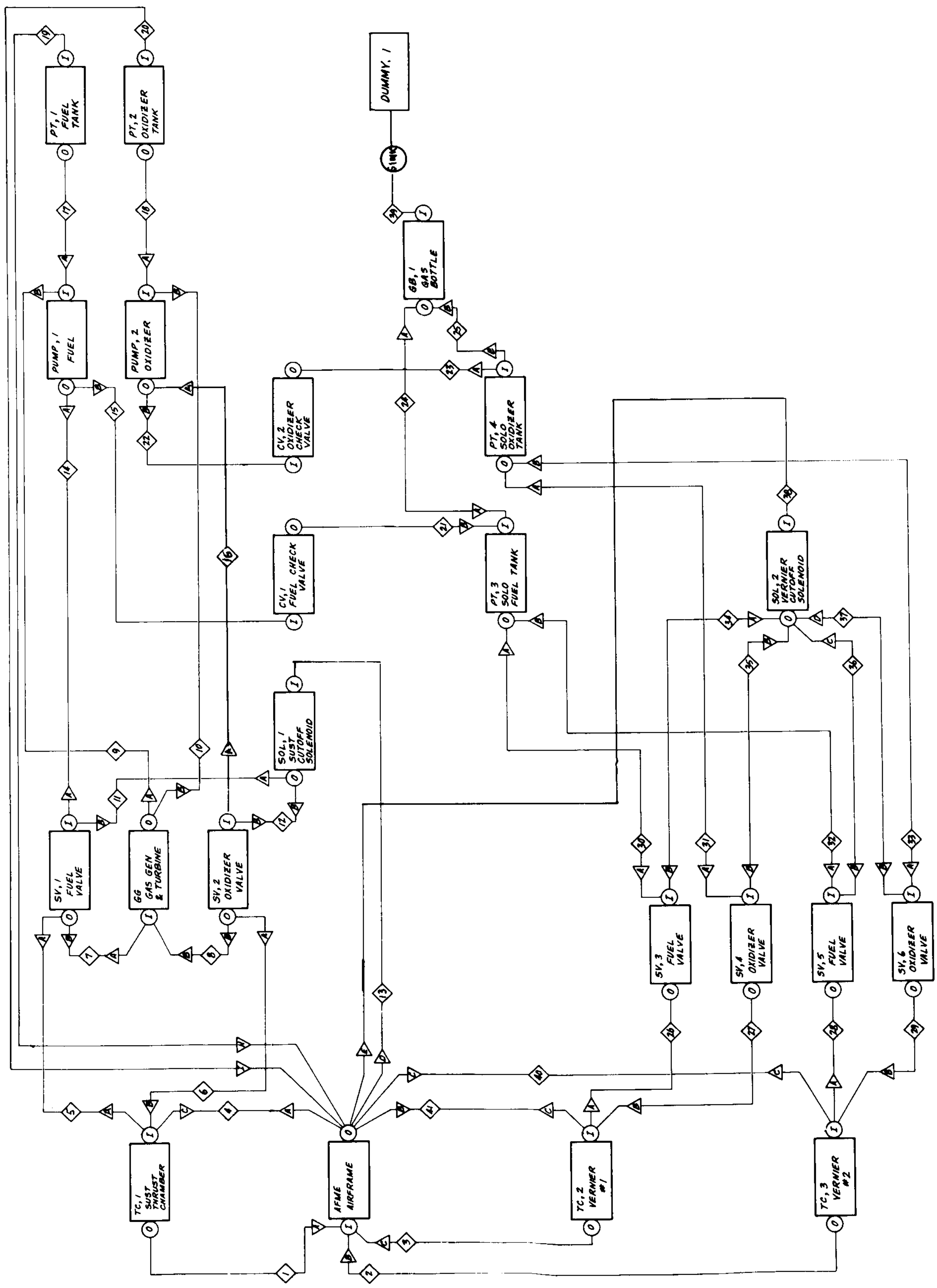


Figure D-9. Modified Sustainer Vehicle Block Diagram

```
IDENT=$SUSTAN$    *
CONNECTIONS.
    AFME(IN.A).TO.TC.1(OUT)
    AFME(IN.B).TO.TC.2(OUT)
    AFME(IN.C).TO.TC.3(OUT)
    AFME(OUT.A).TO.TC.1(IN.C)
    TC.1(IN.A).TO.SV.1(OUT.A)
    TC.1(IN.B).TO.SV.2(OUT.A)
    GG(IN.A).TO.SV.1(OUT.B)
    GG(IN.B).TO.SV.2(OUT.B)
    PUMP.1(IN.B).TO.GG(OUT.A)
    PUMP.2(IN.B).TO.GG(OUT.B)
    SV.1(IN.B).TO.SOL.1(OUT.A)
    SV.2(IN.B).TO.SOL.1(OUT.B)
    SOL.1(IN).TO.AFME(OUT.D)
    SV.1(IN.A).TO.PUMP.1(OUT.A)
    CV.1(IN).TO.PUMP.1(OUT.B)
    SV.2(IN.A).TO.PUMP.2(OUT.A)
    PUMP.1(IN.A).TO.PT.1(OUT)
    PUMP.2(IN.A).TO.PT.2(OUT)
    PT.1(IN).TO.AFME(OUT.H)
    PT.2(IN).TO.AFME(OUT.I)
    PT.3(IN.B).TO.CV.1(OUT)
    CV.2(IN).TO.PUMP.2(OUT.B)
    PT.4(IN.A).TO.CV.2(OUT)
    PT.3(IN.A).TO.GB.1(OUT.A)
    PT.4(IN.B).TO.GB.1(OUT.B)
    TC.2(IN.A).TO.SV.3(OUT)
    TC.2(IN.B).TO.SV.4(OUT)
    TC.3(IN.A).TO.SV.5(OUT)
    TC.3(IN.B).TO.SV.6(OUT)
    SV.3(IN.A).TO.PT.3(OUT.A)
    SV.4(IN.A).TO.PT.4(OUT.A)
    SV.5(IN.A).TO.PT.3(OUT.B)
    SV.6(IN.A).TO.PT.4(OUT.B)
    SV.3(IN.B).TO.SOL.2(OUT.A)
    SV.4(IN.B).TO.SOL.2(OUT.B)
    SV.5(IN.B).TO.SOL.2(OUT.C)
    SV.6(IN.B).TO.SOL.2(OUT.D)
    SOL.2(IN).TO.AFME(OUT.E)
    GB.1(IN).TO.DUMMY.1(SINK)
    TC.3(IN.C).TO.AFME(OUT.C)
    TC.2(IN.C).TO.AFME(OUT.B)


EQUIVALENCE.
    WDF1            =WDL(TC.1(IN.A))
    WDF2            =WDL(TC.2(IN.A))
    WDF3            =WDL(TC.3(IN.A))
    WDO1            =WDL(TC.1(IN.B))
    WDO2            =WDL(TC.2(IN.B))
    WDO3            =WDL(TC.3(IN.B))
```

Figure D-10. Modified Sustainer Vehicle Source Program (1 of 3)

```
INPUT PARAMETERS.
   ARFRL(AFME(IN))
   APRL(AFME(IN))
   CGLOT(AFME(IN))
   CGLOST(AFME(IN))
   CGLFT(AFME(IN))
   CGSFT(AFME(IN))
   VWD(AFME(IN))
   CGLN(AFME(OUT))
   CGY(AFME(OUT))
   CGX(AFME(OUT))
   AIXX(AFME(OUT))
   AIYY(AFME(OUT))
   AIZZ(AFME(OUT))
   SVEL(AFME(IN))
   ATA(AFME(IN))
   PAPP(AFME(IN))
   CC(AFME(IN))
   ZD(AFME(OUT))
   PFPA(AFME(OUT))
   PAA(AFME(OUT))
   CSA(AFME(IN))
   COPP(AFME(IN))
   COPY(AFME(IN))
   CD(AFME(IN))
   CGLM(AFME(IN))
   RD(AFME(OUT))
   TTT(AFME(OUT,D))
   TTT(AFME(OUT,E))
   VC(AFME(OUT,D))
   VC(AFME(OUT,E))
   KCH(TC,2(IN))
   KCH(TC,3(IN))
   KCH(TC,1(IN))
   APA(AFME(OUT,B))
   CP(TC,1(IN))
   CP(TC,2(IN))
   CP(TC,3(IN))
   EXRAT(TC,1(IN))
   EXRAT(TC,2(IN))
   EXRAT(TC,3(IN))
   TAR(TC,1(IN))
   TAR(TC,2(IN))
   TAR(TC,3(IN))
   WDF(TC,1(IN))
   WDF(TC,2(IN))
   WDF(TC,3(IN))
   OFMR(TC,1(IN))
   OFMR(TC,2(IN))
   OFMR(TC,3(IN))
   OFMR(GG(IN))
   WDF(GG(IN,A))
   DWDF(GG(IN,A))
   DRPS(GG(OUT))
```

Figure D-10. Modified Sustainer Vehicle Source Program (2 of 3)

```
     DOL(PT,1(IN))
     DOL(PT,2(IN))
     DOL(PT,3(IN))
     DOL(PT,4(IN))
     WL(PT,1(IN))
     WL(PT,2(IN))
     WL(PT,3(IN))
     WL(PT,4(IN))
     WG(PT,1(IN))
     WG(PT,2(IN))
     WLPR(PUMP,1(IN))
     PR(PUMP,1(OUT))
     WLPR(PUMP,2(IN))
     PR(PUMP,2(OUT))
     PR(GB(OUT))


 DESIRED RESULTS.
     TAT(AFME(OUT))
     WDL(PT,1(OUT))
     WDL(PT,2(OUT))


 NEW ELEMENT TAPE.
$              COMPILE MAD,PRINT OBJECT,PUNCH OBJECT
              STATEMENT LABEL M
              BOOLEAN IFRT
              IFRT=1B
 START        CONTINUE
-             READ AND PRINT DATA
             RFIRST PART OF CALCULATIONS GOES HERE
              TRANSFER TO BEGIN
-     BACK    CONTINUE
             RSECOND PART OF CALCULATIONS GOES HERE
              P'S TIME,X,Y,Z
              TRANSFER TO START
-    BEGIN    CONTINUE
              TRANSFER TO BACK
+             END OF PROGRAM
 CALCULATIONS.
-             CONTINUE
+    MAINLP   CONTINUE
```

Figure D-10. Modified Sustainer Vehicle Source Program (3 of 3)

```
 ELEMENT DESCRIPTION.                                                          0004D
 NAME OF ELEMENT = AFME                                                        0004N
 ATTACHMENT NAMES = IN,OUT                                                     0004A
           BR'S=RPS.
BR'S=TAT.        TISP,TWT,AACC,Q,CSA,CAF,ARFRL,COPP,PCP,COPY ,CP,CD
BR'S=APA,RD,DRAG,CGLM,CGLOT,CGLOST,CGLFT,VWD,OOT,OSOT,FFT,FSFT,CGLN,SVEL,
BR'S=MACH,PAPP,PPDVS,PPV,CC,ZD,PFPA,PAA
 ST'N.
    ISPTC(IN,A),THEN,TISP(OUT)
           S=0.
           S=S+⊕ISPTC(IN,A)⊕
           ⊕TISP(OUT)⊕=S
 ST'N.
    TAT(OUT),TWT(IN),AACC(OUT),Q(IN),CSA(IN),CSA(IN),THEN,CAF(OUT)
           ⊕CAF(OUT)⊕=(⊕TAT(OUT)⊕-(⊕TWT(IN)⊕*⊕AACC(OUT)⊕))/(⊕Q(IN)⊕*⊕CS
          1A(IN)⊕
 ST'N.
    ARFRL(IN),COPP(IN),THEN,PCP(OUT)
           ⊕PCP(OUT)⊕=⊕ARFRL(IN)⊕*⊕COPP(IN)⊕
 ST'N.
    ARFRL(IN),COPY(IN),THEN,YCP(OUT)
           ⊕YCP(OUT)⊕=⊕ARFRL(IN)⊕*⊕COPY(IN)⊕
 ST'N.
    CD(IN),CSA(IN),APA(IN),RD(IN),THEN,DRAG(IN)
           ⊕DRAG(IN)⊕=⊕CD(IN)⊕*⊕CSA(IN)⊕*⊕APA(IN)⊕*⊕ RD(IN)⊕**2*0.5
 ST'N.
    TAT(OUT),TWT(IN),AACC(OUT),THEN,DRAG(IN)
           ⊕DRAG(IN)⊕=⊕TAT(OUT)⊕-⊕TWT(IN)⊕*⊕AACC(OUT)⊕
 ST'N.
    CGLM(IN),CGLOT(IN),CGLOST(IN),CGLFT(IN),CGSFT(IN),VWD(IN),OOT(IN),OS
1   OT(IN),FFT(IN),FSFT(IN),THEN,CGLN(OUT)
           ⊕CGLN(OUT)⊕=(⊕CGLM(IN)⊕*⊕VWD(IN)⊕+⊕CGLOT(IN)⊕*⊕OOT(IN)⊕+⊕CGLO
          1ST(IN)⊕*⊕OSOT(IN)⊕+⊕CGLFT(IN)⊕*⊕FFT(IN)⊕+⊕CGSFT(IN)⊕*⊕FSFT(I
          2N)⊕)/(⊕VWD(IN)⊕+⊕OOT(IN)⊕+⊕OSOT(IN)⊕+⊕FFT(IN)⊕+⊕FSFT(IN)⊕)
 ST'N.
    COMPUTE,CGLN(OUT),CGY(OUT),CGX(OUT),AIXX(OUT),AIYY(OUT),AIZZ(OUT)
           ⊕CGLN(OUT)⊕=CGLN.(MT)
           ⊕CGY(OUT)⊕=CGY.(MT)
           ⊕CGX(OUT)⊕=CGX.(MT)
           ⊕AIXX(OUT)⊕=AIXX.(MT)
           ⊕AIYY(OUT)⊕=AIYY.(MT)
           ⊕AIZZ(OUT)⊕=AIZZ.(MT)
 ST'N.
    PAPP(IN),PPDVS(IN),CC(IN),THEN,PAPP(IN)
           ⊕PPDS(IN)⊕=⊕PPDVS(IN)⊕*⊕PPV(IN)⊕
           ⊕DPA(IN)⊕=⊕PPDS(IN)⊕*⊕CC(IN)⊕
           ⊕PAPP(IN)⊕=⊕PAPP(IN)⊕+⊕DPA(IN)⊕
 ST'N.
    ZD(OUT),RD(OUT),THEN,PFPA(OUT)
           COSF=⊕ZD(OUT)⊕/⊕RD(OUT)⊕
           ⊕ARCOSF(IN)⊕=ARCOSF.(DEG,COSF)
           ⊕PFPA(OUT)⊕=⊕ARCOSF(IN)⊕
 ST'N.
    PFPA(OUT),PAPP(IN),THEN,PAA(OUT)
```

Figure D-11. Modified Sustainer Vehicle Element Descriptions (1 of 6)

```
            ⊕PAA(OUT)⊕=⊕PFPA(OUT)⊕-⊕PAPP(IN)⊕
 ST'N.
    COMPUTE.PAA(OUT)
            ⊕PAA(OUT)⊕=PAA.(MT)
 ST'N.
    TWT(IN).AACC(OUT).DRAG(IN).THEN.TAT(OUT)
            ⊕TAT(OUT)⊕=⊕DRAG(IN)⊕+⊕TWT(IN)⊕*⊕AACC(OUT)⊕
 ST'N.
    AACC(OUT).DRAG(IN).TAT(OUT).THEN.TWT(IN)
            ⊕TWT(IN)⊕=(⊕TAT(OUT)⊕-⊕DRAG(IN)⊕)/⊕AACC(OUT)⊕
 ST'N.
    DRAG(IN).TAT(OUT).TWT(IN).THEN.AACC(OUT)
            ⊕AACC(OUT)⊕=(⊕TAT(OUT)⊕-⊕DRAG(IN)⊕)/⊕TWT(IN)⊕
 ST'N.
    VWD(IN).WL(OUT.A).THEN.TWT(IN)
            S=0.
            S=S+⊕WL(OUT.A)⊕
            ⊕TWT(IN)⊕=⊕VWD(IN)⊕+S
 ST'N.                                                          0004C011
  ATTC(IN.A).THEN.TAT(OUT.A)
            S=0.
            S=S+⊕ATTC(IN.A)⊕
            ⊕TAT(OUT)⊕=S
DS'D.                                                           0010

 ELEMENT DESCRIPTION.                                           0010D
 NAME OF ELEMENT = TC                                           0010N
 ATTACHMENT NAMES = IN.OUT                                      0010A
BR'S=KCH.EXRAT.TAR.OFMR
 ST'N.
    WDF(IN).WDO(IN).THEN.CP(IN)
            W'R ⊕WDF(IN)⊕.E.O..OR.⊕WDO(IN)⊕.E.O.
            ⊕CP(IN)⊕=0.
            O'E
            ⊕CP(IN)⊕=⊕CP(IN)⊕
 ST'N.
    RD(IN).SVEL(IN).THEN.M(IN)
            ⊕M(IN)⊕=⊕RD(IN)⊕/⊕SVEL(IN)⊕
 ST'N.
    M(IN).ATA(IN).THEN.Q(IN)
            ⊕Q(IN)⊕=0.7*⊕ATA(IN)⊕*⊕M(IN)⊕**2
 ST'N.                                                          0010C001
    KCH(IN).APA(IN).CP(IN).EXRAT(IN).TAR(IN).THEN.ATTC(OUT)     0010C001
            ⊕ATTC(OUT)⊕=(⊕KCH(IN)⊕-(⊕ATA(IN)⊕/⊕CP(IN)⊕)*⊕EXRAT(IN)⊕)*⊕CP 0010C001
           1.IN)⊕*⊕TAR(IN)⊕                                     0010C001
 ST'N.                                                          0010C002
 WDO.WDF(IN).ATTC(OUT).THEN.ISPTC(OUT)
            ⊕WDP(IN)⊕=⊕WDO(IN)⊕+⊕WDF(IN)⊕
            ⊕ISPTC(OUT)⊕=⊕ATTC(OUT)⊕/⊕WDP(IN)⊕                  0010C003
 ST'N.                                                          0010C004
    WDF(IN).OFMR(IN).THEN.WDO(IN)                               0010C004
            ⊕WDO(IN)⊕=⊕WDF(IN)⊕*⊕OFMR(IN)⊕                      0010C004
 ST'N.                                                          0010C005
    WDO(IN).OFMR(IN).THEN.WDF(IN)                               0010C005
```

Figure D-11. Modified Sustainer Vehicle Element Descriptions (2 of 6)

```
            ⊕WDF(IN)⊕=⊕WDO(IN)⊕*(1./⊕OFMR(IN)⊕)                                   0010C005
DS'D

 ELEMENT DESCRIPTION.                                                              0011D
 NAME OF ELEMENT = GG                                                              0011N
 ATTACHMENT NAMES = IN.OUT                                                         0011A
BR'S=ATA.DRPS
 ST'N.                                                                             0011C001
    OFMR(IN).WDF(IN).THEN.WDO(IN)                                                  0011C001
            ⊕WDO(IN)⊕=⊕WDF(IN)⊕*⊕OFMR(IN)⊕                                       0011C001
 ST'N.                                                                             0011C002
    WDO(IN).DWDO(IN).WDF(IN).DWDF(IN).DRPS(OUT).THEN.RPS(OUT)                      0011C002
            W'R ⊕WDO(IN)⊕.E.⊕DWDO(IN)⊕.AND.⊕WDF(IN)⊕.E.⊕DWDF(IN)⊕               0011C002
            ⊕RPS(OUT)⊕=⊕DRPS(OUT)⊕                                               0011C002
            O'E                                                                    0011C002
            ⊕RPS(OUT)⊕=F(⊕WDO(IN)⊕.⊕WDF(IN)⊕)                                    0011C002
            E'L                                                                    0011C002
 ST'N.                                                                             0011C003
    DWDF(IN).OFMR(IN).THEN.DWDO(IN).                                               0011C003
            ⊕DWDO(IN)⊕=⊕DWDF(IN)⊕*⊕OFMR(IN)⊕                                     0011C003
DS'D.

 ELEMENT DESCRIPTION.                                                              0012D
 NAME OF ELEMENT = SV                                                              0012N
 ATTACHMENT NAMES =IN.OUT                                                          0012A
BR'S=SM.M.PR.NPR
 ST'N.                                                                             0012C001
    SM(IN).THEN.M(IN)                                                              0012C001
            W'R ⊕SM(IN)⊕.E.0.0                                                    0012C001
            ⊕M(IN)⊕=1.                                                            0012C001
            O'E                                                                    0012C001
            ⊕M(IN)⊕=0.                                                            0012C001
            E'L                                                                    0012C001
 ST'N.                                                                             0012C002
    M(IN).WD(IN).THEN.WD(OUT)                                                      0012C002
            W'R ⊕M(IN)⊕.E.1                                                       0012C002
            ⊕WD(OUT)⊕=⊕WD(IN)⊕                                                   0012C002
            O'E                                                                    0012C002
            ⊕WD(OUT)⊕=0.                                                          0012C002
            E'L                                                                    0012C002
 ST'N.                                                                             0012C003
    M(IN).WD(OUT).THEN.WD(IN)                                                      0012C003
            W'R ⊕M(IN)⊕.E.1.                                                      0012C003
            ⊕WD(IN)⊕=⊕WD(OUT)⊕                                                   0012C003
            O'E                                                                    0012C003
            ⊕WD(IN)⊕=0.0                                                          0012C003
            E'L                                                                    0012C003
 ST'N.                                                                             0012C004
    M(IN).PR(IN).THEN.PR(OUT)                                                      0012C004
            W'R ⊕M(IN)⊕.E.1.                                                      0012C004
            ⊕PR(OUT)⊕=⊕PR(IN)⊕                                                   0012C004
            O'E                                                                    0012C004
            ⊕PR(OUT)⊕=⊕NPR(OUT)⊕                                                 0012C004
            E'L                                                                    0012C004
```

Figure D-11. Modified Sustainer Vehicle Element Descriptions (3 of 6)

```
 ST'N.                                                        0012C005
    PR(OUT).THEN.NPR(OUT)                                     0012C005
            ⊕NPR(OUT)⊕=⊕PR(OUT)⊕                              0012C005
 ST'N.                                                        0012C006
    M(IN).PR(OUT).THEN.PR(IN)                                 0012C006
            W'R ⊕M(IN)⊕.E.1.                                  0012C006
            ⊕PR(IN)⊕=⊕PR(OUT)⊕                                0012C006
            O'E                                               0012C006
            ⊕PR(IN)⊕=⊕NPR(IN)⊕                                0012C006
            E'L                                               0012C006
 ST'N.                                                        0012C007
    PR(IN).THEN.NPR(IN)                                       0012C007
            ⊕NPR(IN)⊕=⊕PR(IN)⊕                                0012C007
DS'D                                                          0012

 ELEMENT DESCRIPTION.                                         0013D
 NAME OF ELEMENT = PUMP                                       0013N
 ATTACHMENT NAMES = IN.OUT                                    0013B
BR'S=RPS.WLPR
 STATEMENT COLLECTION.                                        0013C001
    RPS(IN).WLPR(IN).THEN.WD(OUT)                             0013C001
            ⊕WD(OUT)⊕=⊕WLPR(IN)⊕*⊕RPS(IN)⊕                    0013C001
 S'C                                                          0013C002
    WD(OUT).THEN.WD(IN)                                       0013C002
            ⊕WD(IN)⊕=⊕WD(OUT)⊕                                0013C002
 S'C                                                          0013C003
    RPS(IN).PR(IN).THEN.PR(OUT)                               0013C003
            ⊕PR(OUT)⊕=F(⊕RPS(IN)⊕.⊕PR(IN)⊕)                   0013C003
DS'D.

 ELEMENT DESCRIPTION.                                         0014D
 NAME OF ELEMENT = PT                                         0014N
 ATTACHMENT NAMES = IN.OUT                                    0014A
BR'S=DOL.DOG.VT
 ST'N.                                                        0014C001
    DOL(IN).VL(IN).THEN.WL(IN)                                0014C001
            ⊕WL(IN)⊕=⊕VL(IN)⊕*⊕DOL(IN)⊕                       0014C001
 ST'N.                                                        0014C002
    WL(IN).DOL(IN).THEN.VL(IN)                                0014C002
            ⊕VL(IN)⊕=⊕WL(IN)⊕/⊕DOL(IN)⊕                       0014C002
 ST'N.                                                        0014C003
    DOG(IN).VG(IN).THEN.WG(IN)                                0014C003
            ⊕WG(IN)⊕=⊕VG(IN)⊕*⊕DOG(IN)⊕                       0014C003
 ST'N.                                                        0014C004
    WG(IN).DOG(IN).THEN.VG(IN)                                0014C004
            ⊕VG(IN)⊕=⊕WG(IN)⊕/⊕DOG(IN)⊕                       0014C004
 ST'N.                                                        0014C005
    WL(OUT).NWL(OUT).THEN.WDL(OUT)                            0014C005
            ⊕WDL(OUT)⊕=⊕WL(OUT)⊕-⊕NWL(OUT)⊕                   0014C005
            ⊕WL(OUT)⊕=⊕NWL(OUT)⊕                              0014C005
            ⊕NWL(OUT)⊕=F(TIME)                                0014C005
 ST'N.                                                        0014C006
    WL(IN).NWL(IN).THEN.WDL(IN)                               0014C006
            ⊕WDL(IN)⊕=⊕WL(IN)⊕-⊕NWL(IN)⊕                      0014C006
```

Figure D-11. Modified Sustainer Vehicle Element Descriptions (4 of 6)

```
              ⊕WL(IN)⊕=⊕NWL(IN)⊕                                0014C006
              ⊕NWL(IN)⊕=F(TIME)                                 0014C006
 ST'N.                                                          0014C007
    WDL(OUT),WL(OUT),THEN,NWL(OUT)
              ⊕NWL(OUT)⊕=⊕WL(OUT)⊕-⊕WDL(OUT)⊕                   0014C007
              ⊕WL(OUT)⊕=⊕NWL(OUT)⊕                              0014C007
              ⊕NWL(OUT)⊕=F(TIME)                                0014C007
 ST'N.                                                          0014C008
    VT(IN),VL(IN),THEN,VU(IN)                                   0014C008
              ⊕VU(IN)⊕=⊕VT(IN)⊕-⊕VL(IN)⊕                        0014C008
 ST'N.                                                          0014C009
    VT(IN),DOL(IN),TEMP(IN),IPR(IN),THEN,PR(IN)                 0014C009
              W'R FRT                                           0014C009
              FRT=0B                                            0014C009
              ⊕PR(IN)⊕=⊕IPR(IN)⊕                                0014C009
              O'E                                               0014C009
              ⊕PR(IN)⊕=⊕DOL(IN)⊕*1545.*⊕TEMP(IN)⊕/⊕VT(IN)⊕      0014C009
 ST'N.                                                          0014C010
              E'L                                               0014C010
    COMPUTE,PR(IN)                                              0014C010
              ⊕PR(IN)⊕=PTPR.(MT,TANKN)                          0014C010
DS'D.                                                           0014

 ELEMENT DESCRIPTION.                                           0015D
 NAME OF ELEMENT = CV                                           0015N
 ATTACHMENT NAMES = IN,OUT                                      0015A
 ST'N.                                                          0015C001
              PR(IN),PR(OUT),THEN,M(IN)
              W'R ⊕PR(OUT)⊕.GE.0.9*⊕PR(IN)⊕                     0015C001
              ⊕M(IN)⊕=1.                                        0015C001
              O'E                                               0015C001
              ⊕M(IN)⊕=0.                                        0015C001
              E'L                                               0015C001
 ST'N.                                                          0015C002
    M(IN),PR(IN),THEN,PR(OUT)                                   0015C002
              W'R ⊕M(IN)⊕.E.1.                                  0015C002
              ⊕PR(OUT)⊕=⊕PR(IN)⊕                                0015C002
              O'E                                               0015C002
              ⊕PR(OUT)⊕=⊕NPR(OUT)⊕                              0015C002
              E'L                                               0015C002
 ST'N.                                                          0015C003
    PR(OUT),THEN,NPR(OUT)                                       0015C003
              ⊕NPR(OUT)⊕=⊕PR(OUT)⊕                              0015C003
 ST'N.                                                          0015C004
    M(IN),WD(IN),THEN,WD(OUT)                                   0015C004
              W'R ⊕M(IN)⊕.E.1.                                  0015C004
              ⊕ND(OUT)⊕=⊕WD(IN)⊕                                0015C004
              O'E                                               0015C004
              ⊕WD(OUT)⊕=0.                                      0015C004
              E'L                                               0015C004
 ST'N.                                                          0015C005
    WD(OUT),WD(IN),THEN,M(IN)                                   0015C005
              W'R ⊕WD(OUT)⊕.E.⊕WD(IN)⊕                          0015C005
              ⊕M(IN)⊕=1.                                        0015C005
```

Figure D-11. Modified Sustainer Vehicle Element Descriptions (5 of 6)

```
          O'E                                                   0015C005
          ⊕M(IN)⊕=0.                                            0015C005
          E'L                                                   0015C005
DS'D.

 ELEMENT DESCRIPTION.                                           0016D
 NAME OF ELEMENT = SOL                                          0016N
 ATTACHMENT NAMES = IN,OUT                                      0016A
          BR'S= TTT,VC
 ST'N.                                                          0016C001
    TTT(IN),VC(IN),CP(IN), RD(IN),THEN,SM(OUT)
          W'R ⊕TTT(IN)⊕.LE.⊕MT(IN)⊕.OR.⊕VC(IN)⊕.LE.⊕ RD(IN)⊕.OR.⊕CP(IN)0016C002
         1⊕.E.0.                                                0016C002
          ⊕V(IN)⊕=0.                                            0016C002
          O'E                                                   0016C002
          ⊕V(IN)⊕=26.                                           0016C002
          E'L                                                   0016C002
          W'R⊕V(IN)⊕.GE.26.                                     0016C001
          ⊕SM(OUT)⊕=1.                                          0016C001
          O'E                                                   0016C001
          ⊕SM(OUT)⊕=0.                                          0016C001
          E'L                                                   0016C001
DS'D.

 ELEMENT DESCRIPTION.                                           0017D
 NAME OF ELEMENT = GB                                           0017N
 ATTACHMENT NAMES = IN,OUT                                      0017A
 ST'N.                                                          0017C001
    COMPUTE,PR(OUT)                                             0017C001
          ⊕PR(OUT)⊕=GBPR.(MT,BOTTN)                             0017C001
DS'D.

EL'N
NM'T  = DUMMY
AT'S = SINK
 DESCRIPTION FINISHED.                                          0004F
   NEXT SET OF DATA
```

Figure D-11. Modified Sustainer Vehicle Element Descriptions (6 of 6)

NAME	DEFINITION	UNITS	
AACC	AXIAL ACCELERATION	G	
APA	AMBIENT PRESSURE	PSIA	
ARFRL	AIRFRAME REFERENCE LENGTH	INCHES	NOTE 2
ATA	AMBIENT TEMPERATURE	DEG F	
ATTC	AXIAL THRUST OF THRUST CHAMBER	POUNDS	
CAF	COEFFICIENT OF AXIAL FORCE	(--)	
CC	COMPUTATIONAL CYCLE	SECS	NOTE 6
CD	COEFFICIENT OF DRAG	(--)	
CGLFT	LONGITUDINAL CENTER OF GRAVITY OF FUEL IN FUEL-TANK	INCHES	NOTE 2
CGLM	LONGITUDINAL CENTER OF GRAVITY, DRY MISSILE	INCHES	NOTE 2
CGLN	LONGITUDINAL CENTER GRAVITY, WET	INCHES	NOTE 2
CGLOST	LONGITUDINAL CENTER OF GRAVITY OF OXIDIZER OF SOLO-TANK	INCHES	NOTE 2
CGLOT	LONGITUDINAL CENTER OF GRAVITY OF OXIDIZER IN OX-TANK	INCHES	NOTE 2
CGSFT	LONGITUDINAL CENTER OF GRAVITY OF FUEL IN SOLO-TANK	INCHES	NOTE 2
COPP	CENTER OF PRESSURE, PITCH (AT CENTER LINE)	(--)	
COPY	CENTER OF PRESSURE, YAW (AT CENTER LINE)	(--)	
CSA	VEHICLE CROSS-SECTIONAL AREA	SQ FT	
DOG	DENSITY OF GAS	POUNDS/CU FT	
DOL	DENSITY OF LIQUID	PDS/CU FT	
DRAG	AERODYNAMIC DRAG	POUNDS/SQ FT	
DRPS	DESIRED ROTATION PER SECOND	RPS	
DWDF	DESIRED FUEL FLOW RATE	POUNDS/SEC	
DWDO	DESIRED OXIDIZER FLOW RATE	POUNDS/SEC	
EXRAT	EXPANSION RATIO OF THRUST CHAMBER	(--)	
FFT	WEIGHT OF FUEL IN FUEL TANK	POUNDS	
FSFT	WEIGHT OF FUEL IN SOLO FUEL-TANK	POUNDS	
IPR	INITIAL PRESSURE	PSIA	
ISPTC	SPECIFIC IMPULSE OF THRUST CHAMBER	(--)	
KCH	CONSTANT, THRUST CHAMBER	(--)	
M	MODE	0-1	
MACH	CALCULATED MACH NUMBER	(--)	
NWL	NEW WEIGHT OF LIQUID	POUNDS	
OFMR	OXIDIZER/FUEL MIXTURE RATIO	(--)	
OOT	WEIGHT OF OXIDIZER, OXIDIZER TANK	POUNDS	
OSOT	WEIGHT OF OXIDIZER IN SOLO OX-TANK	POUNDS	
PAA	PITCH ANGLE OF ATTACK	DEG	NOTE 1
PAPP	PITCH ANGLE, PITCH PROGRAM	DEG	NOTE 1
PCP	PITCH CENTER OF PRESSURE (AT CENTER LINE)	INCHES	NOTE 3
PFPA	PROGRAMMED FLIGHT PATH ANGLE	DEG	NOTE 1
PPDVS	PITCH PROGRAMMER DEGREES PER VOLT-SECS	DEG/VOLT-SEC	NOTE 5
PPV	PITCH PROGRAMMER VOLTS OUT	VOLTS	
PR	PRESSURE	PSIA	
Q	DYNAMIC PRESSURE	POUNDS/SQ FT	
RD	RESULTANT VELOCITY	FT/SEC	

Figure D-12. Parameter Descriptions (1 of 2)

NAME	DEFINITION	UNITS	
RPS	ROTATION PER SECOND	RPS	
SM	SOLENOID MODE	0-1	
SVEL	VELOCITY OF SOUND, AT ALTITUDE	FT/SEC	
TAR	THROAT AREA	SQ IN	
TAT	TOTAL AXIAL THRUST	POUNDS	
TEMP	TEMPERATURE, LOCAL, OR OF A MASS	DEG F	
TISP	TOTAL SPECIFIC IMPULSE	(--)	
TTT	THRUST TERMINATION TIME	SECS	
TWT	TOTAL WEIGHT OF VEHICLE	POUNDS	
V	VOLTAGE	VOLTS	
VC	VELOCITY CRITERIA	FT/SEC	NOTE 7
VG	VOLUME OF GAS	CU FT	
VL	VOLUME OF LIQUID	CU FT	
VT	VOLUME OF TANK	CU FT	
VU	ULLAGE VOLUME	CU FT	
VWD	VEHICLE WEIGHT, DRY	POUNDS	NOTE 4
WD	FLOW RATE	POUNDS	
WDF	FUEL FLOW RATE	POUNDS/SEC	
WDL	FLOW RATE OF LIQUID	POUNDS/SEC	
WDO	OXIDIZER FLOW RATE	POUNDS/SEC	
WDP	PROPELLANT FLOW RATE	POUNDS/SEC	
WG	WEIGHT OF GAS	POUNDS	
WL	WEIGHT OF LIQUID, PROPELLANT TANK	POUNDS	
WL	WEIGHT OF LIQUID	POUNDS	
WLPR	WEIGHT LIQUID MOVED PER ROTATION	POUNDS	
WPR	NEW PRESSURE	PSIA	
YCP	YAW CENTER OF PRESSURE (AT CENTER LINE)	INCHES	NOTE 3
ZD	VELOCITY IN Z-PLANE (UP)	FT/SEC	

Note 1. See Figure D-13.

Note 2. Vehicle station numbers, in inches (Figure D-14).

Note 3. Center of pressure is given as a station number, related to the vehicle center line.

Note 4. Vehicle weight (dry) includes all weight except for the propellants in the tanks. Propellants in the propellant lines, hydraulic fluids, etc., are included in the "dry weight" parameter.

Note 5. Flight Control (programmer) is considered to be contained in the AFME for the present analysis.

Note 6. Computational cycle is the "number of looks-per-second" by the computer simulation.

Note 7. Velocity criteria is a preset value in the guidance system (AFME), to command thrust termination when vehicle velocity equals velocity criteria.

Figure D-12. Parameter Descriptions (2 of 2)

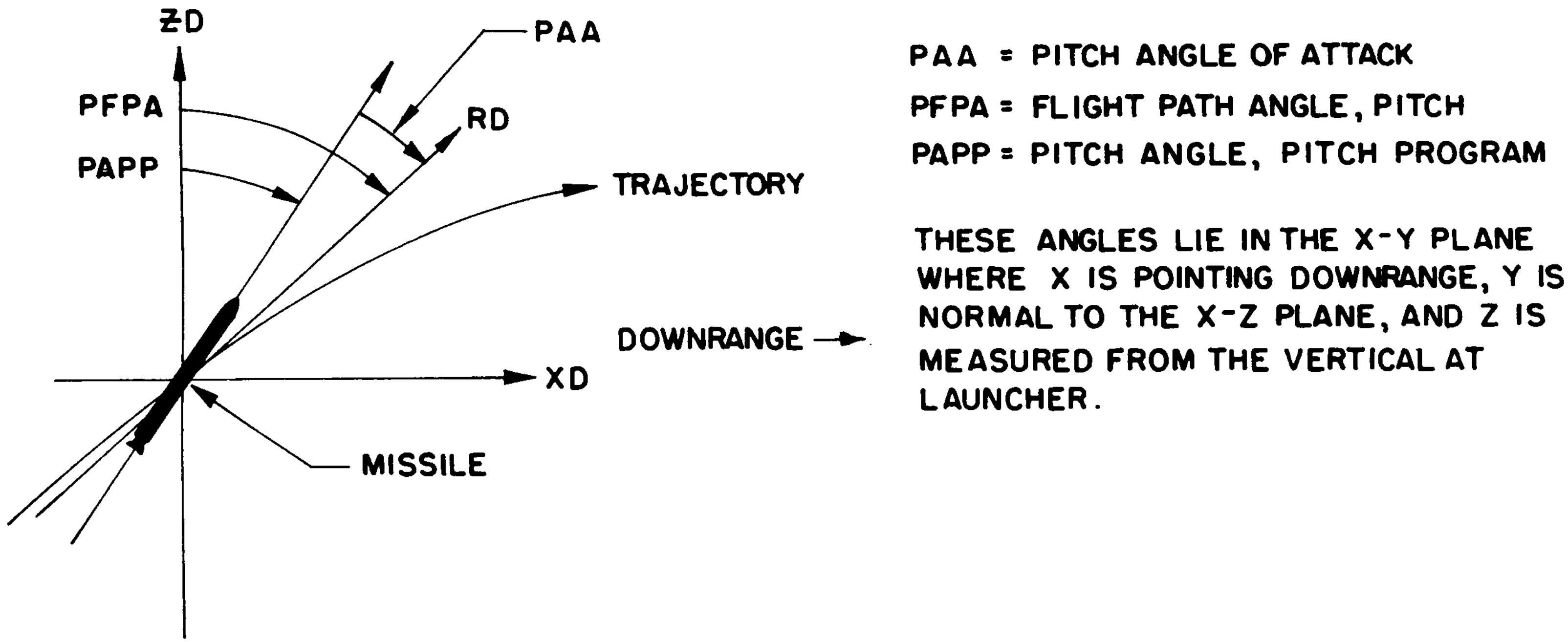


Figure D-13. Pitch Angle Notations

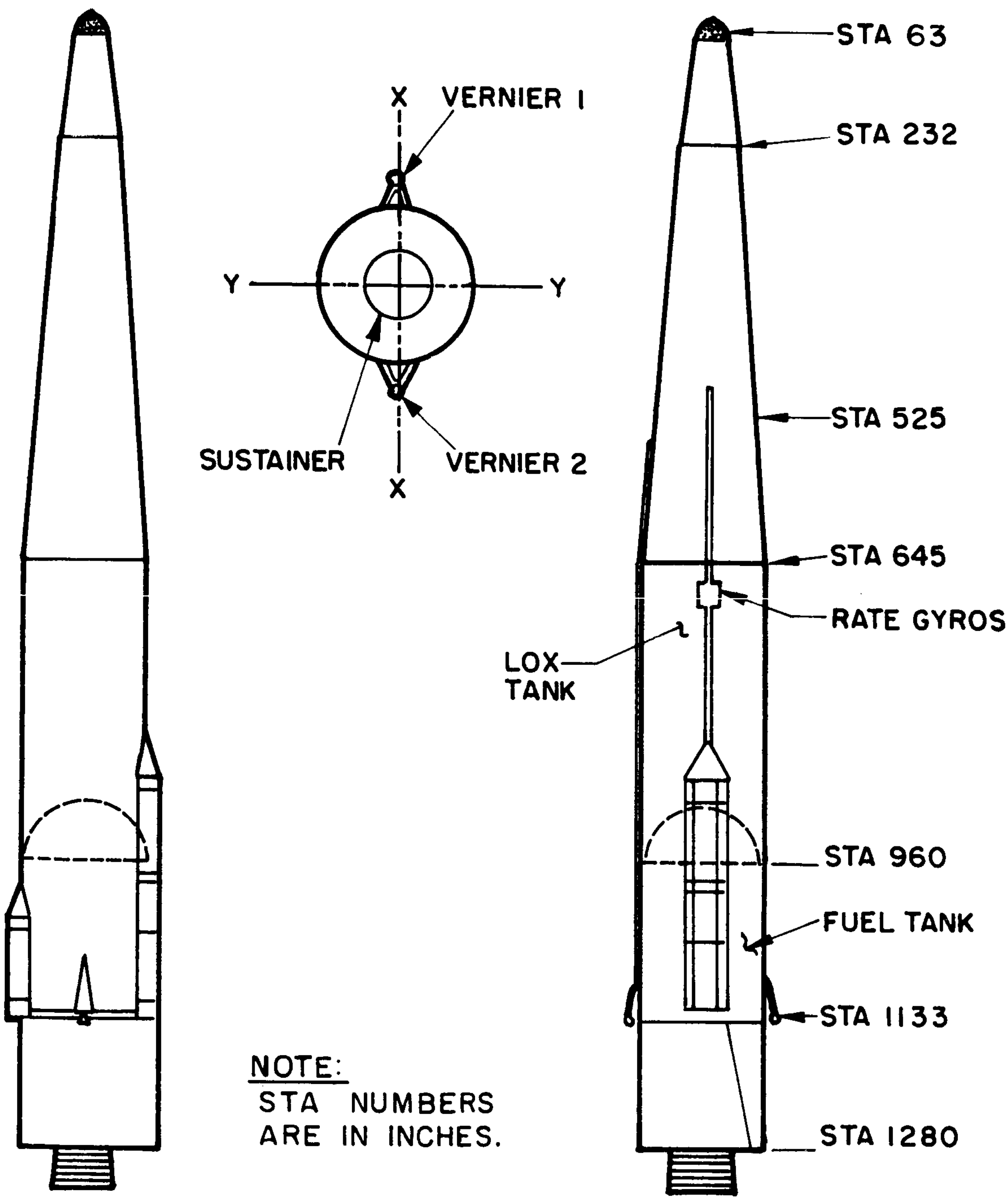


Figure D-14. SLVS Station Identification

4 LAUNCH VEHICLE RELIABILITY

In the reliability simulation phase of the Study Launch Vehicle System Analysis, the Flight Control Subsystem (Section 2, Figure D-3) was partitioned into four equipments;

a. Autopilot,
b. Excitation Transformer,
c. Programmer, and
d. Cabling.

The autopilot at the equipment level was further partitioned into three components. A diagram of the SLVS which shows this augmented system definition appears in Figure D-15. These additional details of the system were included for two reasons;

a. To demonstrate the "nesting" procedure (Appendix B, Section 3. 2) to a greater degree;

b. To show that uniform subsystem levels are not required in the simulation and analysis.

The information required for Reliability Simulation of the SLVS is given by the SDA, the SMA, and numeric data. These facets of the SLVS reliability simulation are discussed in the following sections.

4.1 SYSTEM DEFINITION ARRAY

As indicated in Appendix B, Volume II, a system to be modeled by the Reliability Simulator must be defined by a System Definition Array (SDA) which describes all the possible system states in a shorthand form. This is accomplished by listing the allowed modes for each element of the system in a single-row array, the columns of which represent the respective elements in the system. Each element of the system was defined to have one "failure" mode and one "good" mode. While the Simulator is capable of modeling systems with elements having many more than two modes, this decision was made to keep the modeling relatively simple in order to provide verification of the unified concept.

Each component of the system in Figure D-15 which is comprised of elements at a lower functional level in the system, must be modeled by the Reliability Simulator as a separate "system." The model will

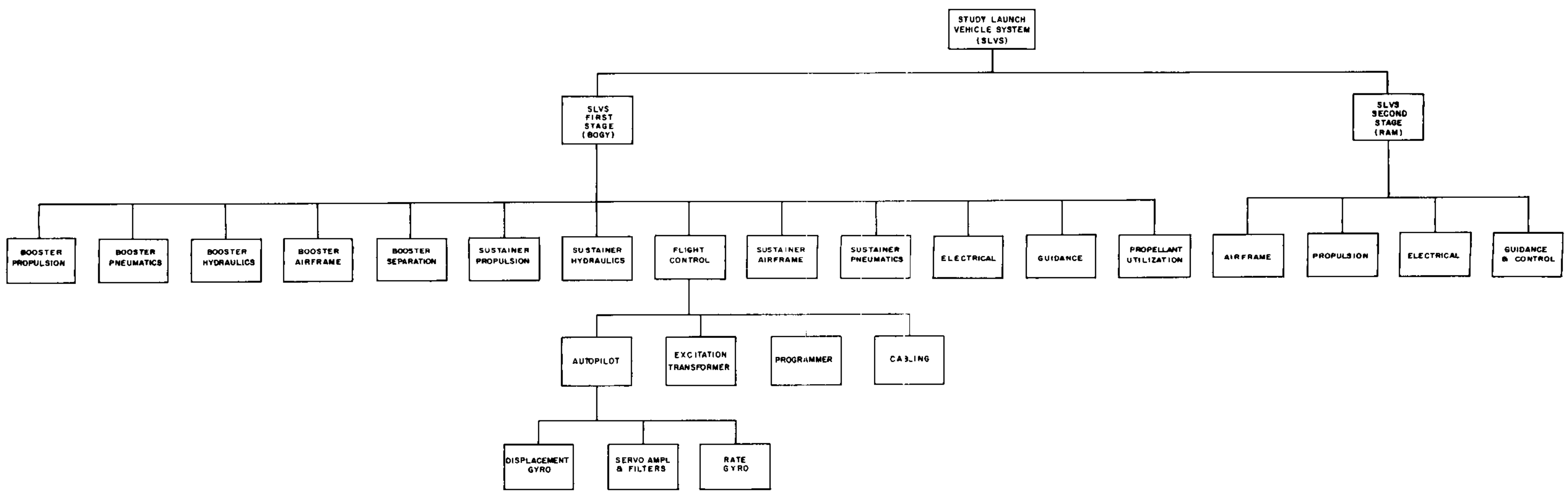


Figure D-15. SLVS Diagram

be in the form of a computer program having the name of the element modeled. The SDA's for the various system elements that must be modeled are given below with the SLVS System Definition Array. The subscript on each SDA entry signifies the element number. Note that as far as the Simulator is concerned, the SLVS is composed of only two elements, the First Stage (element 1) and Second Stage (element 2); both of these require further modeling from information supplied for elements at lower system levels. While the element numbers are sequential starting at "1" for all the "systems," the elements associated with the symbol may differ among systems. For example, element "1" for "SLVS First Stage" is the "Booster Propulsion" system, while element "1" for "1ST. STAGE FLIGHT CONTROL" is the "FLT. CONT. CABLING."

Functional Level*	SDA
SYSTEM	$1_1 1_2$
SLVS FIRST STAGE	$1_1 1_2 1_3 1_4 1_5 1_6 1_7 1_8 1_9 1_{10} 1_{11} 1_{12} 1_{13}$
1ST. STG. FLT. CONTROL	$1_1 1_2 1_3 1_4$
FLT. CONT. AUTOPILOT	$1_1 1_2 1_3$
SLVS SECOND STAGE	$1_1 1_2 1_3 1_4$

4.2 SYSTEM MODE ARRAY

In addition to the SDA, which describes all the possible system states, another array (the Systems Mode Array, SMA) is required to describe the subset of these states which constitutes the system mode to be modeled.

*Defined by the indentation depth.

It was determined early in the SLVS definition phase, that the reliability simulation would be restricted to modeling the probability of only one mode for the SLVS, which would be called the "failure" mode. The system in Figure D-15 is referred to as a "series system," since failure of any single component represented by a block in the figure will result in the system reaching a failure state. Thus, the rows (submodes) of the SMA for each "system" will be of order one. The SMA's corresponding to the respective "system" SDA's are given below. The supermode, Δ_1, is used in the same manner as described in Appendix B (Volume II) Section 2.1.1.

SLVS

$1_1 \Delta_2$

$\Delta_1 1_2$

SLVS FIRST STAGE

$1_1\Delta_2\Delta_3\Delta_4\Delta_5\Delta_6\Delta_7\Delta_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1 1_2\Delta_3\Delta_4\Delta_5\Delta_6\Delta_7\Delta_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2 1_3\Delta_4\Delta_5\Delta_6\Delta_7\Delta_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3 1_4\Delta_5\Delta_6\Delta_7\Delta_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4 1_5\Delta_6\Delta_7\Delta_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5 1_6\Delta_7\Delta_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5\Delta_6 1_7\Delta_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5\Delta_6\Delta_7 1_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5\Delta_6\Delta_7\Delta_8 1_9\Delta_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5\Delta_6\Delta_7\Delta_8\Delta_9 1_{10}\Delta_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5\Delta_6\Delta_7\Delta_8\Delta_9\Delta_{10} 1_{11}\Delta_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5\Delta_6\Delta_7\Delta_8\Delta_9\Delta_{10}\Delta_{11} 1_{12}\Delta_{13}$

$\Delta_1\Delta_2\Delta_3\Delta_4\Delta_5\Delta_6\Delta_7\Delta_8\Delta_9\Delta_{10}\Delta_{11}\Delta_{12} 1_{13}$

1ST STG. FLIGHT CONTROL

$1_1\Delta_2\Delta_3\Delta_4$

$\Delta_1 1_2\Delta_3\Delta_4$

$\Delta_1\Delta_2 1_3\Delta_4$

$\Delta_1\Delta_2\Delta_3 1_4$

FLIGHT CONTROL AUTOPILOT

$1_1\Delta_2\Delta_3$

$\Delta_1 1_2\Delta_3$

$\Delta_1\Delta_2 1_3$

SLVS SECOND STAGE

$1_1\Delta_2\Delta_3\Delta_4$

$\Delta_1 1_2\Delta_3\Delta_4$

$\Delta_1\Delta_2 1_3\Delta_4$

$\Delta_1\Delta_2\Delta_3 1_4$

4.3 DATA

Since reliability is a time dependent parameter, the SLVS time sequence, or mission, has been defined as shown in Table D-I. The Sequence of Events tabulation contains only "major" event/time relations.

Unclassified flight data for the Atlas system served as the basis for the SLVS first stage reliability data. Since these were flight data,

Table D-I. Reliability Time Base

Abbreviated Sequence of Events

BOGY/RAM

Time		Event	Function
Seconds	Hours		
0.0	0.00000	Lift-off (two-inch motion)	Programmer
135.	.03750	Staging Discrete Command	Guidance
136.	.03778	Booster Engines Cutoff	Programmer
138.	.03833	Jettison Booster Section	Programmer
264.	.07347	Sustainer Engine Cutoff	Guidance
280.	.07772	Vernier Engines Cutoff	Guidance
282.	.07839	Separation Discrete Command	Guidance
322.	.08951	Fire Ullage Rockets	RAM G &C
332.	.09222	Initiate RAM Burn	RAM G &C
449.	.12476	RAM Engine Cutoff	RAM G &C

N. B. - The events/time entries above are only those necessary for use in the flight reliability modeling and are not intended to be representative for performance analysis.

each of the element environmental stress factors (K factor) for the first stage elements is "1" during the active state of the element. After the point in time where the element function is no longer needed, the associated K factor goes to zero, signifying that an element failure would not be recognized during that time. The unclassified data which served as the basis for the SLVS second stage reliability data set were available only in generic form. Therefore, the K factors for the second stage components are much larger than those for the first stage components. Table D-II presents the stress factors for each element of the SLVS.

The SDA and SMA are the basic information needed in order for the Reliability Simulator to automatically generate a system model. However, in order to compute system probabilities, information must be available concerning the element failure rates and the admissible modes of the system elements.

These types of data exist in varied forms for equipment similar to the SLVS components. A failure rate journal was developed from available data, relating an "element failure rate" to each element of the SLVS which required the exponential failure rate model. The pertinent reliability data from the journal which follows were recorded on the Data Transmittal Forms from which input tabulation cards were prepared.

4.3.1 MTBF DATA INPUTS

The following notes are applicable to the listings in Figure D-16.

a. The MTBF DATA INPUTS is a catalog listing of manufacturer's data, showing Reliability Function Block identification number (RFB), component name, and MTBF.

b. The MTBF listing is used where failure rates are not available.

c. The MTBF DATA INPUTS are processed to derive the adjusted failure rate inputs to the FAILURE RATE JOURNAL.

d. In the current listing, some of the entries have been postulated, where manufacturer's data were not available.

e. These MTBF values are for flight environment.

Table D-II. SLVS Environmental Factors

ELEMENT	TIME INTERVAL								
	0.0000 to .0375	.0375 to .03778	.03778 to .03833	.03833 to .07347	.07347 to .07772	.07772 to .07839	.07839 to .08951	.08951 to .09222	.09222 to .12476
	K FACTORS								
BOOSTER PROPULSION	1.	1.	0.	0.	0.	0.	0.	0.	0.
BOOSTER PNEUMATICS	1.	1.	0.	0.	0.	0.	0.	0.	0.
BOOSTER HYDRAULICS	1.	1.	0.	0.	0.	0.	0.	0.	0.
BOOSTER AIRFRAME	1.	1.	1.	0.	0.	0.	0.	0.	0.
BOOSTER SEPARATION	1.	1.	1.	0.	0.	0.	0.	0.	0.
SUSTAINER PROPULSION	1.	1.	1.	1.	1.	0.	0.	0.	0.
SUSTAINER HYDRAULICS	1.	1.	1.	1.	1.	0.	0.	0.	0.
SUSTAINER AIRFRAME	1.	1.	1.	1.	1.	1.	0.	0.	0.
SUSTAINER PNEUMATICS	1.	1.	1.	1.	1.	1.	0.	0.	0.
PROPELLANT UTILIZATION	1.	1.	1.	1.	0.	0.	0.	0.	0.
1ST STG. ELECTRICAL	1.	1.	1.	1.	1.	1.	0.	0.	0.
1ST STG. GUIDANCE	1.	1.	1.	1.	1.	1.	0.	0.	0.
1ST STG. FLIGHT CONTROL	-	-	-	-	-	-	-	-	-
FLIGHT CONTROL CABLING	1.	1.	1.	1.	1.	1.	0.	0.	0.
FLIGHT CONTROL EXC. TRANSFORMER	1.	1.	1.	1.	1.	0.	0.	0.	0.
FLIGHT CONTROL PROGRAMMER	1.	1.	1.	1.	1.	1.	0.	0.	0.
FLIGHT CONTROL AUTOPILOT	-	-	-	-	-	-	-	-	-
AUTOPILOT DISP. GYRO	1.	1.	1.	1.	1.	0.	0.	0.	0.
AUTOPILOT SERVO AMP. & FILTERS	1.	1.	1.	1.	1.	0.	0.	0.	0.
AUTOPILOT RATE GYRO	1.	1.	1.	1.	1.	0.	0.	0.	0.
2ND STG. AIRFRAME	200.	100.	100.	100.	50.	1.	1.	50.	200.
2ND STG. PROPULSION	200.	100.	100.	100.	50.	1.	1.	50.	200.
2ND STG. ELECTRICAL	200.	100.	100.	100.	50.	1.	1.	50.	200.
2ND STG. GUIDANCE & CONTROL	1.	1.	1.	1.	1.	1.	1.	1.	1.

4.3.2 FAILURE RATE JOURNAL

The FAILURE RATE JOURNAL notes which apply to Figure D-17 are:

a. The FAILURE RATE JOURNAL is a catalog listing of reliability information encompassing component RFB number, component name, source of data, the applicable environment, and the failure rates for upper, mean, and lower values. The failure rates are given per million hours.

b. The FAILURE RATE JOURNAL is used as a library for the ELEMENT FAILURE RATES, where the element failure rates are identified with the SLVS under consideration.

c. The mean failure rates cited are based upon realistic data.

d. The upper and lower failure rate values are synthetic, not to be taken as realistic, and are included only for demonstration purposes. These values should not be relied upon for any attempt at true evaluation of the system.

4.3.3 ELEMENT FAILURE RATES

The following notes refer to the listing of ELEMENT FAILURE RATES:

a. The ELEMENT FAILURE RATES (Figure D-18*) is a catalog listing of the mean, upper, and lower failure rates for each element of the SLVS. Failure rates for the components of each element are listed by code, and the total failure rate is given for each element.

b. The mean values of ELEMENT FAILURE RATES are based upon realistic data, but the upper and lower values are arbitrary.

---

*The individual elements shown in Figure D-18 are ordinarily printed on separate pages, however, to conserve space, they have been combined.

****
BASED ON
MANUFACTURER'S DATA
****
STAGE 1 - BOGY
****

RELIABILITY FUNCTION BLOCK			COMPONENT	MTBF
20	0		VEHICLE AIRFRAME	
20	1		SUSTAINER SECTION	968.
20	2		BOOSTER SECTION	672.
21	0		FLIGHT CONTROL	
21	1		PROGRAMMER	37.6
21	2		AUTOPILOT	
21	2	2	DISPL GYRO PACKAGE	11.6
21	2	3	SERVO AMPLS FILTERS	42.1
21	2	4	RATE GYRO PACKAGE	59.6
21	3		EXCITATION XFMR	242.8
21	4		CABLING	375.5
23	0		VEHICLE PROP SYS	
23	5		SUST VERNIER SUBSYS	
23	5	1	SUSTAINER CHAMBER	29.6
23	5	2	SUST FUEL PRE-VALVE	112.0
23	5	3	VERNIER CHAMBERS	1115.
23	5	4	FUEL PLUMBING	1001.2
23	5	6	MAIN LOX PRE-VALVE	34.5
23	5	8	MAIN LOX STAG VALVE	23.3
23	5	10	LOX PB AND FITTINGS	1501.9
23	5	12	VERN LOX PB AND FIT	600.7
23	5	14	FUEL FILL CK VALV PB	3003.7
23	6		BOOSTER SUBSYSTEM	
23	6	1	BOOSTER CHAMBERS	21.4
23	6	2	FUEL PREVALVE	112.0
23	6	4	MAIN FUEL STAGING VLV	116.1
23	6	6	FUEL PB AND FITTINGS	3003.7
23	6	8	LOX PB AND FITTINGS	3003.7
23	6	10	LOX A/B F AND D VALVE	28.3
23	6	12	FUEL A/B F AND D VALVE	112.0
24	0		VEHICLE SEPARATION SYS	
24	1		BOOSTER SEPARATION SYS	
24	1	2	EXPLOSIVE VALVES	196.7
24	1	4	HE DIST MANIFOLD	1230.
24	1	6	FITTINGS 10	156.
24	1	8	JETT TRACK INSTALL	620.
24	1	10	SLIDE INSTALL	470.
24	1	12	SUPPLY LINE	146.
25	0		VEHICLE ELECT SYSTEM	
25	1	1	MAIN 28 VDC BATTERY	54.9
25	2		115 VAC 400 CPS ROT IN	12.9
25	4		CABLING	300.4
25	5		DISTRIBUTION BOX	296.4
25	6		JUNCTION BOX	312.2

Figure D-16. MTBF Data Inputs (1 of 2)

26	0			VLH PNEU AND HYD SYS	
26	1			PNEUMATIC SYSTEM	
26	1	1		SUSTAINER PNEU SYSTEM	
26	1	1	4	LOX BOIL-OFF VALVE	30.8
26	1	1	6	LOX TANK PRESS REG	924.5
26	1	1	8	LOX TANK PRESS REL VLV	924.5
26	1	1	11	LOX TANK PRESS MAN SOV	3003.7
26	1	1	15	FUEL TANK PRESS MAN	3003.7
26	1	1	20	AMB HE STORAGE BTL	3003.7
26	1	1	24	STAGING DISC'S	1001.2
26	1	1	26	LINE AND FITTINGS	3003.7
26	1	1	30	CABLING AND WIRING	3003.7
26	1	2		BOOSTER PNEU SYSTEM	
26	1	2	4	FUEL TANK PRESS REG	924.5
26	1	2	5	FUEL TANK PRESS REL VA	924.5
26	1	2	16	AMB HEAT EXCHGR	3003.7
26	1	2	18	HE STORAGE BOTTLES 6	3003.7
26	1	2	20	AMB HE STOR BTL SEP	3003.7
26	1	2	26	LINES 7 FITTINGS	3003.7
26	1	2	32	CABLING AND WIRING	3003.7
26	2	1		BOOSTER HYD SYSTEM	
26	2	1	2	HYD FLUID TANK	3003.7
26	2	1	4	HYDRAULIC PUMP	3.3
26	2	1	6	ACCUMULATOR	36.2
26	2	1	7	MAIN DIST MANIFOLD	3003.7
26	2	1	8	SUPPLY MANIFOLD	3003.7
26	2	1	9	RETURN MANIFOLD	3003.8
26	2	1	10	BOOSTER ACTUATOR ASSEM	60.3
26	2	1	12	PB VALVES AND FILTERS	750.9
26	2	1	14	RISE-OFF DISC	3003.7
26	2	1	16	HYD BLEED FITTINGS	
26	2	1	18	CABLING	3003.7
26	2	2		SUST HYD SYSTEM	
26	2	2	2	HYD FLUID TANK	3003.7
26	2	2	4	HYDRAULIC PUMP	6.4
26	2	2	8	SUPPLY MANIFOLD	3003.7
26	2	2	9	RETURN MANIFOLD	3003.7
26	2	2	10	SUST ACTUATOR ASSEM	68.4
26	2	2	16	STAGING DISC-SUST HALF	3003.7
26	2	2	17	STAG DISC BOOST HALF	3003.7
26	2	2	18	VERNIER/SOLO ACCUM MAN	3003.7
26	2	2	20	INBD VERN ACTUATOR	26.1
26	2	2	21	OUTBD VERN ACTUATOR	45.0
26	2	2	22	VERN/SOLO ACCUM	320.8
26	2	2	24	PB VALVES AND FILTERS	3003.7
26	2	2	26	RISE-OFF DISC	3003.7
26	2	2	28	CABLING	3003.7
27	0			PROP UTIL AND LEV CONT	
27	1			PROP UTIL SYSTEM	
27	1	2		COMPUTER ASSEMBLY	1680.
27	1	4		LOX STILLWELL ASSEMBLY	3003.7
27	1	6		FULL STILLWELL ASSY	3003.7

Figure D-16. MTBF Data Inputs (2 of 2)

- -

FAILURE RATE JOURNAL

- -

RFB ELEMENT

* * * * * * * * * * *

9 20 1 SUSTAINER AIRFRAME

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	4890.
MEAN VALUE	1033.
LOWER EXTREME	904.

4 20 2 BOOSTER AIRFRAME

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	6117.
MEAN VALUE	1488.
LOWER EXTREME	1334.

3 21 1 PROGRAMMER

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	46166.
MEAN VALUE	26596.
LOWER EXTREME	25943.

Figure D-17. Failure Rate Journal (1 of 30)

- -

FAILURE RATE JOURNAL

- -

```
1 21  2  2     DISPL GYRO PACKAGE
                                          SOURCE     --     RIPPR 12-1-63

                                          ENVIRONMENT = FLIGHT TRIAL

                                          FAILURE RATE
                                               UPPER EXTREME          121440.
                                               MEAN VALUE              86207.
                                               LOWER EXTREME           85032.


2 21  2  3     SERVO AMPLS FILTERS
                                          SOURCE     --     RIPPR 12-1-63

                                          ENVIRONMENT = FLIGHT TRIAL

                                          FAILURE RATE
                                               UPPER EXTREME           42247.
                                               MEAN VALUE              23753.
                                               LOWER EXTREME           23136.


3 21  2  4     RATE GYRO PACKAGE
                                          SOURCE     --     RIPPR 12-1-63

                                          ENVIRONMENT = FLIGHT TRIAL

                                          FAILURE RATE
                                               UPPER EXTREME           32322.
                                               MEAN VALUE              16779.
                                               LOWER EXTREME           16260.


2 21  3        EXCITATION XFMR
                                          SOURCE     --     RIPPR 12-1-63

                                          ENVIRONMENT = FLIGHT TRIAL

                                          FAILURE RATE
                                               UPPER EXTREME           11820.
                                               MEAN VALUE               4119.
                                               LOWER EXTREME            3862.
```

Figure D-17. Failure Rate Journal (2 of 30)

FAILURE RATE JOURNAL

1 21 4 CABLING

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	8856.
MEAN VALUE	2663.
LOWER EXTREME	2457.

6 23 5 1 SUSTAINER CHAMBER

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	55840.
MEAN VALUE	33784.
LOWER EXTREME	33049.

6 23 5 2 SUST FUEL PRE-VALVE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	20268.
MEAN VALUE	8929.
LOWER EXTREME	8551.

6 23 5 3 VERNIER CHAMBERS

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	4491.
MEAN VALUE	897.
LOWER EXTREME	777.

Figure D-17. Failure Rate Journal (3 of 30)

6 23 5 4 FUEL PLUMBING

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	4791.
MEAN VALUE	999.
LOWER EXTREME	872.

6 23 5 6 MAIN LOX PRE-VALVE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	49416.
MEAN VALUE	28986.
LOWER EXTREME	28305.

6 23 5 8 MAIN LOX STAG VALVE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	67779.
MEAN VALUE	42918.
LOWER EXTREME	42090.

6 23 5 10 LOX PB AND FITTINGS

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	3762.
MEAN VALUE	666.
LOWER EXTREME	563.

Figure D-17. Failure Rate Journal (4 of 30)

FAILURE RATE JOURNAL

6 23 5 12 VERN LOX PB AND FIT

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	6561.
MEAN VALUE	1665.
LOWER EXTREME	1502.

6 23 5 14 FUEL FILL CK VALV PB

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

1 23 6 1 BOOSTER CHAMBERS

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	72669.
MEAN VALUE	46729.
LOWER EXTREME	45864.

1 23 6 2 FUEL PREVALVE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	20268.
MEAN VALUE	8929.
LOWER EXTREME	8551.

Figure D-17. Failure Rate Journal (5 of 30)

FAILURE RATE JOURNAL

1 23 6 4 MAIN FUEL STAGING VLV

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	19750.
MEAN VALUE	8613.
LOWER EXTREME	8242.

1 23 6 6 FUEL PB AND FITTINGS

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

1 23 6 8 LOX PB AND FITTINGS

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

1 23 6 10 LOX A/B F AND D VALVE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	57893.
MEAN VALUE	35336.
LOWER EXTREME	34584.

Figure D-17. Failure Rate Journal (6 of 30)

FAILURE RATE JOURNAL

1 23 6 12 FUEL A/B F AND D VALVE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	20268.
MEAN VALUE	8929.
LOWER EXTREME	8551.

5 24 1 2 EXPLOSIVE VALVES

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	13640.
MEAN VALUE	5084.
LOWER EXTREME	4799.

5 24 1 4 HE DIST MANIFOLD

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	4235.
MEAN VALUE	813.
LOWER EXTREME	699.

5 24 1 6 FITTINGS 10

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	16018.
MEAN VALUE	6410.
LOWER EXTREME	6090.

Figure D-17. Failure Rate Journal (7 of 30)

FAILURE RATE JOURNAL

---

5 24 1 8 JETT TRACK INSTALL

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	6432.
MEAN VALUE	1613.
LOWER EXTREME	1452.

5 24 1 10 SLIDE INSTALL

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	7663.
MEAN VALUE	2128.
LOWER EXTREME	1943.

5 24 1 12 SUPPLY LINE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	16781.
MEAN VALUE	6849.
LOWER EXTREME	6518.

11 25 1 1 MAIN 28 VDC BATTERY

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	34410.
MEAN VALUE	18215.
LOWER EXTREME	17675.

Figure D-17. Failure Rate Journal (8 of 30)

FAILURE RATE JOURNAL

11 25 2 115 VAC 400 CPS ROT IN

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	110930.
MEAN VALUE	77519.
LOWER EXTREME	76406.

11 25 4 CABLING

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	10252.
MEAN VALUE	3329.
LOWER EXTREME	3098.

11 25 5 DISTRIBUTION BOX

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	10344.
MEAN VALUE	3374.
LOWER EXTREME	3141.

11 25 6 JUNCTION BOX

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	9995.
MEAN VALUE	3203.
LOWER EXTREME	2977.

Figure D-17. Failure Rate Journal (9 of 30)

- -

FAILURE RATE JOURNAL

- -

10 26 1 1 4 LOX BOIL-OFF VALVE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	54090.
MEAN VALUE	32468.
LOWER EXTREME	31747.

10 26 1 1 6 LOX TANK PRESS REG

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	5028.
MEAN VALUE	1082.
LOWER EXTREME	950.

10 26 1 1 8 LOX TANK PRESS REL VLV

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	5028.
MEAN VALUE	1082.
LOWER EXTREME	950.

10 26 1 1 11 LOX TANK PRESS MAN SOV

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

Figure D-17. Failure Rate Journal (10 of 30)

FAILURE RATE JOURNAL

```
10 26  1  1 15 FUEL TANK PRESS MAN
                                        SOURCE      --     RIPPR 12-1-63

                                        ENVIRONMENT = FLIGHT TRIAL

                                        FAILURE RATE
                                             UPPER EXTREME              2522.
                                             MEAN VALUE                  333.
                                             LOWER EXTREME               260.


10 26  1  1 20 AMB HE STORAGE BTL
                                        SOURCE      --     RIPPR 12-1-63

                                        ENVIRONMENT = FLIGHT TRIAL

                                        FAILURE RATE
                                             UPPER EXTREME              2522.
                                             MEAN VALUE                  333.
                                             LOWER EXTREME               260.


10 26  1  1 24 STAGING DISC'S
                                        SOURCE      --     RIPPR 12-1-63

                                        ENVIRONMENT = FLIGHT TRIAL

                                        FAILURE RATE
                                             UPPER EXTREME              4791.
                                             MEAN VALUE                  999.
                                             LOWER EXTREME               872.


10 26  1  1 26 LINE AND FITTINGS
                                        SOURCE      --     RIPPR 12-1-63

                                        ENVIRONMENT = FLIGHT TRIAL

                                        FAILURE RATE
                                             UPPER EXTREME              2522.
                                             MEAN VALUE                  333.
                                             LOWER EXTREME               260.
```

Figure D-17. Failure Rate Journal (11 of 30)

FAILURE RATE JOURNAL

10 26 1 1 30 CABLING AND WIRING

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

2 26 1 2 4 FUEL TANK PRESS REG

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	5028.
MEAN VALUE	1082.
LOWER EXTREME	950.

2 26 1 2 5 FUEL TANK PRESS REL VA

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	5028.
MEAN VALUE	1082.
LOWER EXTREME	950.

2 26 1 2 16 AMB HEAT EXCHGR

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

Figure D-17. Failure Rate Journal (12 of 30)

FAILURE RATE JOURNAL

```
2 26  1  2 18 HE STORAGE BOTTLES 6
                                        SOURCE     --     RIPPR 12-1-63

                                        ENVIRONMENT = FLIGHT TRIAL

                                        FAILURE RATE
                                             UPPER EXTREME            2522.
                                             MEAN VALUE                333.
                                             LOWER EXTREME             260.


2 26  1  2 20 AMB HE STOR BTL SEP
                                        SOURCE     --     RIPPR 12-1-63

                                        ENVIRONMENT = FLIGHT TRIAL

                                        FAILURE RATE
                                             UPPER EXTREME            2522.
                                             MEAN VALUE                333.
                                             LOWER EXTREME             260.


2 26  1  2 26 LINES 7 FITTINGS
                                        SOURCE     --     RIPPR 12-1-63

                                        ENVIRONMENT = FLIGHT TRIAL

                                        FAILURE RATE
                                             UPPER EXTREME            2522.
                                             MEAN VALUE                333.
                                             LOWER EXTREME             260.


2 26  1  2 32 CABLING AND WIRING
                                        SOURCE     --     RIPPR 12-1-63

                                        ENVIRONMENT = FLIGHT TRIAL

                                        FAILURE RATE
                                             UPPER EXTREME            2522.
                                             MEAN VALUE                333.
                                             LOWER EXTREME             260.
```

Figure D-17. Failure Rate Journal (13 of 30)

FAILURE RATE JOURNAL

3 26 2 1 2 HYD FLUID TANK

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

3 26 2 1 4 HYDRAULIC PUMP

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	369088.
MEAN VALUE	303030.
LOWER EXTREME	300828.

3 26 2 1 6 ACCUMULATOR

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	47569.
MEAN VALUE	27624.
LOWER EXTREME	26959.

3 26 2 1 7 MAIN DIST MANIFOLD

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

Figure D-17. Failure Rate Journal (14 of 30)

FAILURE RATE JOURNAL

3 26 2 1 8 SUPPLY MANIFOLD

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

3 26 2 1 9 RETURN MANIFOLD

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

3 26 2 1 10 BOOSTER ACTUATOR ASSEM

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	32037.
MEAN VALUE	16584.
LOWER EXTREME	16069.

3 26 2 1 12 PB VALVES AND FILTERS

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	5711.
MEAN VALUE	1332.
LOWER EXTREME	1186.

Figure D-17. Failure Rate Journal (15 of 30)

FAILURE RATE JOURNAL

---

3 26 2 1 14 RISE-OFF DISC

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

3 26 2 1 18 CABLING

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

7 26 2 2 2 HYD FLUID TANK

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

7 26 2 2 4 HYDRAULIC PUMP

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	203684.
MEAN VALUE	156250.
LOWER EXTREME	154669.

Figure D-17. Failure Rate Journal (16 of 30)

- -

FAILURE RATE JOURNAL

- -

7 26 2 2 8 SUPPLY MANIFOLD

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

7 26 2 2 9 RETURN MANIFOLD

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

7 26 2 2 10 SUST ACTUATOR ASSEM

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	29129.
MEAN VALUE	14620.
LOWER EXTREME	14136.

7 26 2 2 16 STAGING DISC-SUST HALF

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

Figure D-17. Failure Rate Journal (17 of 30)

- -

FAILURE RATE JOURNAL

- -

7 26 2 2 17 STAG DISC BOOST HALF

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

7 26 2 2 18 VERNIER/SOLO ACCUM MAN

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

7 26 2 2 20 INBD VERN ACTUATOR

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	61803.
MEAN VALUE	38314.
LOWER EXTREME	37531.

7 26 2 2 21 OUTBD VERN ACTUATOR

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	40111.
MEAN VALUE	22222.
LOWER EXTREME	21626.

Figure D-17. Failure Rate Journal (18 of 30)

- -

FAILURE RATE JOURNAL

- -

7 26 2 2 22 VERN/SOLO ACCUM

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	9817.
MEAN VALUE	3117.
LOWER EXTREME	2894.

7 26 2 2 24 PB VALVES AND FILTERS

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

7 26 2 2 26 RISE-OFF DISC

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

7 26 2 2 28 CABLING

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE	
UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

Figure D-17. Failure Rate Journal (19 of 30)

FAILURE RATE JOURNAL

---

8 27 1 4 LOX STILLWELL ASSEMBLY

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

8 27 1 6 FUEL STILLWELL ASSY

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	2522.
MEAN VALUE	333.
LOWER EXTREME	260.

8 27 1 2 COMPUTER ASSEMBLY

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	3523.
MEAN VALUE	595.
LOWER EXTREME	498.

12 GUIDANCE

SOURCE -- RIPPR 12-1-63

ENVIRONMENT = FLIGHT TRIAL

FAILURE RATE

UPPER EXTREME	3873.
MEAN VALUE	699.
LOWER EXTREME	594.

Figure D-17. Failure Rate Journal (20 of 30)

- -

FAILURE RATE JOURNAL

- -

RFB ELEMENT

* * * * * * * * * * *

1 1 1 STRUCTURE

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	521.
MEAN VALUE	59.
LOWER EXTREME	28.

1 1 2 PIN PULLERS

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	1361.
MEAN VALUE	308.
LOWER EXTREME	238.

2 2 1 ENGINE

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	1879.
MEAN VALUE	516.
LOWER EXTREME	425.

Figure D-17. Failure Rate Journal (21 of 30)

FAILURE RATE JOURNAL

2 2 2	ULLAGE ROCKETS	
	SOURCE -- UNSPECIFIED	
	ENVIRONMENT = BENCH	
	FAILURE RATE	
	UPPER EXTREME	1602.
	MEAN VALUE	401.
	LOWER EXTREME	321.
2 2 3	BSTR RETRO	
	SOURCE -- UNSPECIFIED	
	ENVIRONMENT = BENCH	
	FAILURE RATE	
	UPPER EXTREME	534.
	MEAN VALUE	62.
	LOWER EXTREME	31.
2 2 5	RETRO MANEUVER ROCKET	
	SOURCE -- UNSPECIFIED	
	ENVIRONMENT = BENCH	
	FAILURE RATE	
	UPPER EXTREME	1602.
	MEAN VALUE	401.
	LOWER EXTREME	321.
2 2 6	PROP FEED/LOAD	
	SOURCE -- UNSPECIFIED	
	ENVIRONMENT = BENCH	
	FAILURE RATE	
	UPPER EXTREME	494.
	MEAN VALUE	54.
	LOWER EXTREME	24.

Figure D-17. Failure Rate Journal (22 of 30)

- -

FAILURE RATE JOURNAL

- -

2 2 7 PROP PRESSURIZATION

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	642.
MEAN VALUE	86.
LOWER EXTREME	49.

2 2 8 HE BOTTLES

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	427.
MEAN VALUE	41.
LOWER EXTREME	16.

2 2 9 HE VALVE SQUIBS

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	435.
MEAN VALUE	43.
LOWER EXTREME	17.

3 3 1 PRI BATTERY

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	425.
MEAN VALUE	41.
LOWER EXTREME	15.

Figure D-17. Failure Rate Journal (23 of 30)

3 3 2	SEC BATTERY	SOURCE -- UNSPECIFIED			
		ENVIRONMENT = BENCH			
		FAILURE RATE			
			UPPER EXTREME		479.
			MEAN VALUE		51.
			LOWER EXTREME		22.
3 3 3	TRANSFER SWITCH	SOURCE -- UNSPECIFIED			
		ENVIRONMENT = BENCH			
		FAILURE RATE			
			UPPER EXTREME		431.
			MEAN VALUE		42.
			LOWER EXTREME		16.
3 3 4	FIL TRANSFORMER	SOURCE -- UNSPECIFIED			
		ENVIRONMENT = BENCH			
		FAILURE RATE			
			UPPER EXTREME		410.
			MEAN VALUE		38.
			LOWER EXTREME		14.
3 3 5	POWER SUPPLY	SOURCE -- UNSPECIFIED			
		ENVIRONMENT = BENCH			
		FAILURE RATE			
			UPPER EXTREME		368.
			MEAN VALUE		31.
			LOWER EXTREME		9.

Figure D-17. Failure Rate Journal (24 of 30)

FAILURE RATE JOURNAL

---

3 3 6 INVERTER

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE

UPPER EXTREME	442.
MEAN VALUE	44.
LOWER EXTREME	17.

3 3 7 FWD J-BOX

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE

UPPER EXTREME	485.
MEAN VALUE	52.
LOWER EXTREME	23.

3 3 8 AFT J-BOX

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE

UPPER EXTREME	464.
MEAN VALUE	48.
LOWER EXTREME	20.

3 3 9 SEP J-BOX

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE

UPPER EXTREME	371.
MEAN VALUE	32.
LOWER EXTREME	9.

Figure D-17. Failure Rate Journal (25 of 30)

FAILURE RATE JOURNAL

3 3 10 DIODE ASSEMBLY

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	368.
MEAN VALUE	31.
LOWER EXTREME	9.

3 3 11 RELAY SWITCH

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	387.
MEAN VALUE	35.
LOWER EXTREME	11.

3 3 12 CONNECTOR

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	385.
MEAN VALUE	34.
LOWER EXTREME	11.

3 3 13 MISC CONN.

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	901.
MEAN VALUE	155.
LOWER EXTREME	105.

Figure D-17. Failure Rate Journal (26 of 30)

FAILURE RATE JOURNAL

4	4	1	IRP		
				SOURCE -- UNSPECIFIED	
				ENVIRONMENT = BENCH	
				FAILURE RATE	
				UPPER EXTREME	1749.
				MEAN VALUE	461.
				LOWER EXTREME	375.
4	4	2	HORIZON SENSOR		
				SOURCE -- UNSPECIFIED	
				ENVIRONMENT = BENCH	
				FAILURE RATE	
				UPPER EXTREME	1149.
				MEAN VALUE	233.
				LOWER EXTREME	172.
4	4	3	VEL METER		
				SOURCE -- UNSPECIFIED	
				ENVIRONMENT = BENCH	
				FAILURE RATE	
				UPPER EXTREME	2074.
				MEAN VALUE	602.
				LOWER EXTREME	504.
4	4	4	F C ELECTRONICS		
				SOURCE -- UNSPECIFIED	
				ENVIRONMENT = BENCH	
				FAILURE RATE	
				UPPER EXTREME	1323.
				MEAN VALUE	294.
				LOWER EXTREME	225.

Figure D-17. Failure Rate Journal (27 of 30)

FAILURE RATE JOURNAL

4	4	5	PRIMARY TIMER		
				SOURCE -- UNSPECIFIED	
				ENVIRONMENT = BENCH	
				FAILURE RATE	
				UPPER EXTREME	844.
				MEAN VALUE	138.
				LOWER EXTREME	91.
4	4	6	SECONDARY TIMER		
				SOURCE -- UNSPECIFIED	
				ENVIRONMENT = BENCH	
				FAILURE RATE	
				UPPER EXTREME	844.
				MEAN VALUE	138.
				LOWER EXTREME	91.
4	4	7	HYD PACKAGE		
				SOURCE -- UNSPECIFIED	
				ENVIRONMENT = BENCH	
				FAILURE RATE	
				UPPER EXTREME	52000.
				MEAN VALUE	40000.
				LOWER EXTREME	39200.
4	4	8	HYD ACTUATOR		
				SOURCE -- UNSPECIFIED	
				ENVIRONMENT = BENCH	
				FAILURE RATE	
				UPPER EXTREME	4195.
				MEAN VALUE	1712.
				LOWER EXTREME	1546.

Figure D-17. Failure Rate Journal (28 of 30)

FAILURE RATE JOURNAL

4 4 9	PNEU VALVE		
		SOURCE -- UNSPECIFIED	
		ENVIRONMENT = BENCH	
		FAILURE RATE	
		UPPER EXTREME	466.
		MEAN VALUE	48.
		LOWER EXTREME	21.
4 4 10	PNEU THRUST CONT		
		SOURCE -- UNSPECIFIED	
		ENVIRONMENT = BENCH	
		FAILURE RATE	
		UPPER EXTREME	588.
		MEAN VALUE	74.
		LOWER EXTREME	39.
4 4 11	PNEU CONTROL PACK		
		SOURCE -- UNSPECIFIED	
		ENVIRONMENT = BENCH	
		FAILURE RATE	
		UPPER EXTREME	1140.
		MEAN VALUE	230.
		LOWER EXTREME	169.
4 4 12	SUN DETECTOR		
		SOURCE -- UNSPECIFIED	
		ENVIRONMENT = BENCH	
		FAILURE RATE	
		UPPER EXTREME	417.
		MEAN VALUE	40.
		LOWER EXTREME	14.

Figure D-17. Failure Rate Journal (29 of 30)

- -

FAILURE RATE JOURNAL

- -

4 4 13 J-BOX,PRI

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	685.
MEAN VALUE	96.
LOWER EXTREME	57.

4 4 14 J-BOX,SEC

SOURCE -- UNSPECIFIED

ENVIRONMENT = BENCH

FAILURE RATE	
UPPER EXTREME	549.
MEAN VALUE	65.
LOWER EXTREME	33.

Figure D-17. Failure Rate Journal (30 of 30)

BOGY

1 BOOSTER PROPULSION

	FAILURE RATES UPPER	MEAN	LOWER
23 6 1	72669.	46729.	45864.
23 6 2	20268.	8929.	8551.
23 6 4	19750.	8613.	8242.
23 6 6	2522.	333.	260.
23 6 8	2522.	333.	260.
23 6 10	57893.	35336.	34584.
23 6 12	20268.	8929.	8551.
SUM =	195892.	109201.	106311.

2 BOOSTER PNEUMATICS

	FAILURE RATES UPPER	MEAN	LOWER
26 1 2 4	5028.	1082.	950.
26 1 2 5	5028.	1082.	950.
26 1 2 16	2522.	333.	260.
26 1 2 18	2522.	333.	260.
26 1 2 20	2522.	333.	260.
26 1 2 26	2522.	333.	260.
26 1 2 32	2522.	333.	260.
SUM =	22669.	3828.	3200.

3 BOOSTER HYDRAULICS

	FAILURE RATES UPPER	MEAN	LOWER
26 2 1 2	2522.	333.	260.
26 2 1 4	369088.	303030.	300828.
26 2 1 6	47569.	27624.	26959.
26 2 1 7	2522.	333.	260.
26 2 1 8	2522.	333.	260.
26 2 1 9	2522.	333.	260.
26 2 1 10	32037.	16584.	16069.
26 2 1 12	5711.	1332.	1186.
26 2 1 14	2522.	333.	260.
26 2 1 18	2522.	333.	260.
SUM =	469539.	350567.	346602.

Figure D-18. SLVS Model-Element Failure Rate (1 of 6)

10 PROPELLANT UTILIZATION

				FAILURE RATES	
			UPPER	MEAN	LOWER
27	1	2	3523.	595.	498.
27	1	4	2522.	333.	260.
27	1	6	2522.	333.	260.
SUM =			8568.	1261.	1017.

11 ELECTRICAL

				FAILURE RATES	
			UPPER	MEAN	LOWER
25	1	1	34410.	18215.	17675.
25	2		110930.	77519.	76406.
25	4		10252.	3329.	3098.
25	5		10344.	3374.	3141.
25	6		9995.	3203.	2977.
SUM =			175932.	105640.	103297.

12 GUIDANCE

		FAILURE RATES	
	UPPER	MEAN	LOWER
	3873.	699.	594.
SUM =	3873.	699.	594.

13 FLIGHT CONTROL

				FAILURE RATES	
			UPPER	MEAN	LOWER
21	4		8856.	2663.	2457.
21	3		11820.	4119.	3862.
21	1		46166.	26596.	25943.
21	2	2	121440.	86207.	85032.
21	2	3	42247.	23753.	23137.
21	2	4	32322.	16779.	16260.
SUM =			262851.	160116.	156691.

Figure D-18. SLVS Model-Element Failure Rate (4 of 6)

RAM

1 S.S. A - AIRFRAME, RAM

		FAILURE RATES		
		UPPER	MEAN	LOWER
1	1	520.8	59.2	28.4
1	2	1361.0	308.0	237.8
	SUM =	1881.8	367.2	266.2

2 S.S. B - PROPULSION, RAM

		FAILURE RATES		
		UPPER	MEAN	LOWER
2	1	1878.9	516.0	425.1
2	2	1602.5	401.0	320.9
2	3	534.4	62.0	30.5
2	5	1602.5	401.0	320.9
2	6	493.9	53.8	24.5
2	7	642.0	85.9	48.8
2	8	427.5	41.4	15.7
2	9	435.3	42.8	16.6
	SUM =	7617.0	1603.9	1203.0

Figure D-18. SLVS Model-Element Failure Rate (5 of 6)

RAM

3 S.S. C - ELECTRICAL, RAM

	FAILURE RATES UPPER	MEAN	LOWER
3 1	425.2	41.0	15.4
3 2	479.5	51.0	22.4
3 3	430.8	42.0	16.1
3 4	410.2	38.4	13.6
3 5	367.6	31.4	9.0
3 6	442.0	44.0	17.5
3 7	484.7	52.0	23.2
3 8	463.7	48.0	20.3
3 9	371.4	32.0	9.4
3 10	367.6	31.4	9.0
3 11	386.9	34.5	11.0
3 12	385.1	34.2	10.8
3 13	901.3	154.8	105.0
SUM =	5916.0	634.7	282.6

4 S.S. D - GUID/CONTL, RAM

	FAILURE RATES UPPER	MEAN	LOWER
4 1	1749.3	461.0	375.1
4 2	1148.9	233.0	171.9
4 3	2074.1	602.0	503.9
4 4	1322.8	294.0	225.4
4 5	843.9	138.3	91.3
4 6	843.9	138.3	91.3
4 7	52000.0	40000.0	39200.0
4 8	4194.6	1712.0	1546.5
4 9	465.8	48.4	20.6
4 10	587.9	73.5	39.2
4 11	1139.9	230.0	169.3
4 12	417.2	39.6	14.4
4 13	684.7	96.2	57.0
4 14	548.7	65.0	32.8
SUM =	68021.7	44131.3	42538.6

Figure D-18. SLVS Model-Element Failure Rate (6 of 6)

5 LAUNCH VEHICLE ECONOMICS

In order to apply the PRESTO concept to a launch vehicle system, models must be developed describing the system performance, reliability, and economics in terms of certain independent parameters of the system. In addition, some of the parameters must be common to all of the models in order that optimization of one of these three system attributes can be performed against certain constraints on the others. Due to difficulties discussed in this final report in producing a working performance simulator, no performance considerations could be made in the unified concept verification and no performance parameters appear in the mathematical models. The system economics simulation was thus limited to only cost-reliability considerations.

5.1 LAUNCH VEHICLE COST CONSIDERATIONS

The approach taken in modeling the SLVS system economics was to determine a cost function which relates cost to failure rate for every basic element. The cost of a system element which is made up of elements at lower system functional levels is determined by summing the costs of the constituent elements.

Section 5.2.1.3 of Volume I of this final report discusses the development of the mathematical model used for the element cost functions. The equation is repeated below.

$$i^{th}\ \text{Element Cost} = \frac{A_i}{\lambda^{B_i}} + C_i \quad (\lambda = \text{failure rate})$$

This model has three parameters that must be determined in order to uniquely define the function for a given element. Three sets of data were determined relating a minimum, nominal, and maximum element failure rate to the corresponding costs. The minimum cost thus determined at the maximum element failure rate was the value of "C" for the element. Simultaneous solution of the two equations resulting from substituting the minimum and nominal points into the above equation, provided unique values of "A" and "B" for each element.

The cost of a system element is obviously affected by a multitude of parameters. In order to facilitate the collection of element cost data, these parameters were "lumped" into categories which can be associated with cost expenditures at various levels in the cycle from "build" to "use" in the launch vehicle. These categories and their definitions are given below.

CATEGORIES

Hardware Cost
Assembly Cost
Shipping to Launch Site Cost
Inspection and Acceptance Cost
Integration Cost
Checkout Cost

DEFINITIONS

Hardware Cost - The sum of all costs accrued to the time an element is manufactured and deemed acceptable.

Assembly Cost - The cost associated in mating a given element with the rest of the system.

Ship-to-Site Cost - The total cost incurred in moving the element from the contractor' s facility to the assembly site of the system.

Inspection and Acceptance Cost - The cost incurred at the assembly site as a result of a need to assure the integrity of the element.

Integration Costs - The element cost resulting from the need to assure that all the individual elements are operating together as a complete system. A typical integration cost would be that associated with assuring that a specific command signal generated within the guidance system does in fact reach the proper point in the propulsion subsystem, and that it does not appear elsewhere. A large portion of these costs are accrued "on the pad."

Checkout Cost - The cost of interfacing the launch vehicle elements and the ground equipment.

5.2 DATA

A COST JOURNAL was developed from available data, establishing a cost value for each category.

5.2.1 COST JOURNAL

Figure D-19 constitutes the COST JOURNAL, with the following applicable notes:

- The COST JOURNAL is a listing of the costs associated with the mean failure rates as given in the Element Failure Rates list.

- The cost values given for the mean failure rates cannot be supported by documentation, and should not be relied upon for any true analysis of system cost.

- The ratio of cost values shown, (such as "hardware cost vs. total cost") were generated largely through the use of trade publications. No serious attempt has been made to certify the validity of these ratios.

- The values given in the COST JOURNAL entries are in dollars.

5.2.2 <u>ELEMENT COST Vs. FAILURE RATES Schedule</u>

The three sets of data for each element, required to define the element cost function were recorded in an Element Cost vs. Failure Rate Schedule (Figure D-20). The following notes are applicable.

- The ELEMENT COST vs. FAILURE RATES schedule is a listing of each system element, showing the three failure rate values, and provision for the two cost factors of Hardware and total Element Cost.

- The COST JOURNAL provides six element cost categories, however, only the Hardware factor and the total Element Cost are included in the ELEMENT COST vs. FAILURE RATES schedule. This was done for simplicity.

- The cost values entered for the upper and lower extremes on failure rates were arbitrarily derived from the nominal cost values, and are not necessarily realistic.

1	BOOSTER PROPULSION		
		HARDWARE	902805.
		ASSEMBLE	361621.
		SHIP TO SITE	37409.
		INSPECT/ACCEPT	289297.
		INTEGRATE	361621.
		CHECKOUT AND LAUNCH	541184.
		ELEMENT COST =	2493936.
2	BOOSTER PNEUMATICS		
		HARDWARE	26984.
		ASSEMBLE	10808.
		SHIP TO SITE	1118.
		INSPECT/ACCEPT	8647.
		INTEGRATE	10808.
		CHECKOUT AND LAUNCH	16175.
		ELEMENT COST =	74541.
3	BOOSTER HYDRAULICS		
		HARDWARE	39265.
		ASSEMBLE	15728.
		SHIP TO SITE	1627.
		INSPECT/ACCEPT	12582.
		INTEGRATE	15728.
		CHECKOUT AND LAUNCH	23538.
		ELEMENT COST =	108468.
4	BOOSTER AIRFRAME		
		HARDWARE	91270.
		ASSEMBLE	36558.
		SHIP TO SITE	3782.
		INSPECT/ACCEPT	29247.
		INTEGRATE	36558.
		CHECKOUT AND LAUNCH	54711.
		ELEMENT COST =	252126.
5	BOOSTER SEPARATION		
		HARDWARE	27726.
		ASSEMBLE	11106.
		SHIP TO SITE	1149.
		INSPECT/ACCEPT	8884.
		INTEGRATE	11106.
		CHECKOUT AND LAUNCH	16620.
		ELEMENT COST =	76590.
6	SUSTAINER PROPULSION		
		HARDWARE	601870.
		ASSEMBLE	241080.
		SHIP TO SITE	24939.
		INSPECT/ACCEPT	192864.
		INTEGRATE	241080.
		CHECKOUT AND LAUNCH	360789.
		ELEMENT COST =	1662624.

Figure D-19. SLVS Journal, Costs For Mean Failure Rate (Sheet 1 of 4)

7	SUSTAINER HYDRAULICS	
	HARDWARE	26177.
	ASSEMBLE	10485.
	SHIP TO SITE	1085.
	INSPECT/ACCEPT	8388.
	INTEGRATE	10485.
	CHECKOUT AND LAUNCH	15692.
	ELEMENT COST =	72312.
8	SUSTAINER AIRFRAME	
	HARDWARE	162257.
	ASSEMBLE	64992.
	SHIP TO SITE	6723.
	INSPECT/ACCEPT	51994.
	INTEGRATE	64992.
	CHECKOUT AND LAUNCH	97265.
	ELEMENT COST =	448224.
9	SUSTAINER PNEUMATICS	
	HARDWARE	42205.
	ASSEMBLE	16905.
	SHIP TO SITE	1749.
	INSPECT/ACCEPT	13524.
	INTEGRATE	16905.
	CHECKOUT AND LAUNCH	25300.
	ELEMENT COST =	116589.
10	PROPELLANT UTILIZATION	
	HARDWARE	33970.
	ASSEMBLE	13607.
	SHIP TO SITE	1408.
	INSPECT/ACCEPT	10885.
	INTEGRATE	13607.
	CHECKOUT AND LAUNCH	20363.
	ELEMENT COST =	93840.
11	ELECTRICAL	
	HARDWARE	50456.
	ASSEMBLE	20210.
	SHIP TO SITE	2091.
	INSPECT/ACCEPT	16168.
	INTEGRATE	20210.
	CHECKOUT AND LAUNCH	30245.
	ELEMENT COST =	139380.
12	GUIDANCE	
	HARDWARE	203321.
	ASSEMBLE	81441.
	SHIP TO SITE	8425.
	INSPECT/ACCEPT	65153.
	INTEGRATE	81441.
	CHECKOUT AND LAUNCH	121880.
	ELEMENT COST =	561660.

Figure D-19. SLVS Journal, Costs For Mean Failure Rate (Sheet 2 of 4)

1	FC CABLING	
	HARDWARE	6664.
	ASSEMBLE	2669.
	SHIP TO SITE	276.
	INSPECT/ACCEPT	2135.
	INTEGRATE	2669.
	CHECKOUT AND LAUNCH	3995.
	ELEMENT COST =	18409.
2	FC EXCITATION TRANSF.	
	HARDWARE	113.
	ASSEMBLE	45.
	SHIP TO SITE	5.
	INSPECT/ACCEPT	36.
	INTEGRATE	45.
	CHECKOUT AND LAUNCH	68.
	ELEMENT COST =	312.
3	FC PROGRAMMER	
	HARDWARE	21908.
	ASSEMBLE	8775.
	SHIP TO SITE	908.
	INSPECT/ACCEPT	7020.
	INTEGRATE	8775.
	CHECKOUT AND LAUNCH	13132.
	ELEMENT COST =	60518.
1	AP DISPLACEMENT GYRO	
	HARDWARE	84730.
	ASSEMBLE	33939.
	SHIP TO SITE	3511.
	INSPECT/ACCEPT	27151.
	INTEGRATE	33939.
	CHECKOUT AND LAUNCH	50791.
	ELEMENT COST =	234062.
2	AP SERVO AMPL/FILTERS	
	HARDWARE	104527.
	ASSEMBLE	41869.
	SHIP TO SITE	4331.
	INSPECT/ACCEPT	33495.
	INTEGRATE	41869.
	CHECKOUT AND LAUNCH	62659.
	ELEMENT COST =	288749.
3	AP RATE GYRO	
	HARDWARE	74700.
	ASSEMBLE	29921.
	SHIP TO SITE	3095.
	INSPECT/ACCEPT	23937.
	INTEGRATE	29921.
	CHECKOUT AND LAUNCH	44779.
	ELEMENT COST =	206354.

Figure D-19. SLVS Journal, Costs For Mean Failure Rate (Sheet 3 of 4)

1	S.S. A - AIRFRAME, RAM	
	HARDWARE	115498.
	ASSEMBLE	46263.
	SHIP TO SITE	4786.
	INSPECT/ACCEPT	37010.
	INTEGRATE	46263.
	CHECKOUT AND LAUNCH	69235.
	ELEMENT COST =	319056.
2	S.S. B - PROPULSION, RAM	
	HARDWARE	690725.
	ASSEMBLE	276672.
	SHIP TO SITE	28621.
	INSPECT/ACCEPT	221337.
	INTEGRATE	276672.
	CHECKOUT AND LAUNCH	414053.
	ELEMENT COST =	1908080.
3	S.S. C - ELECTRICAL, RAM	
	HARDWARE	23779.
	ASSEMBLE	9525.
	SHIP TO SITE	985.
	INSPECT/ACCEPT	7620.
	INTEGRATE	9525.
	CHECKOUT AND LAUNCH	14254.
	ELEMENT COST =	65688.
4	S.S. D - GUID/CONTL, RAM	
	HARDWARE	302334.
	ASSEMBLE	121101.
	SHIP TO SITE	12528.
	INSPECT/ACCEPT	96880.
	INTEGRATE	121101.
	CHECKOUT AND LAUNCH	181233.
	ELEMENT COST =	835176.

Figure D-19. SLVS Journal, Costs For Mean Failure Rate (Sheet 4 of 4)

1 BOOSTER PROPULSION

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	195892.	451403.	1246968.
MEAN VALUE	109201.	902805.	2493936.
LOWER EXTREME	106311.	1406345.	3884932.

2 BOOSTER PNEUMATICS

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	22669.	13492.	37270.
MEAN VALUE	3828.	26984.	74541.
LOWER EXTREME	3200.	39257.	108444.

3 BOOSTER HYDRAULICS

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	469539.	18633.	54234.
MEAN VALUE	350567.	39265.	108468.
LOWER EXTREME	346602.	61383.	130123.

4 BOOSTER AIRFRAME

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	6117.	45635.	126063.
MEAN VALUE	1488.	91270.	252126.
LOWER EXTREME	1334.	168464.	465371.

5 BOOSTER SEPARATION

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	64768.	13863.	38295.
MEAN VALUE	22897.	27726.	76590.
LOWER EXTREME	21501.	37235.	102860.

Figure D-20. Element Cost Vs. Failure Rate
(Sheet 1 of 5)

6 SUSTAINER PROPULSION

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	215429.	300935.	831312.
MEAN VALUE	119176.	601870.	1662624.
LOWER EXTREME	115967.	1036000.	2861876.

7 SUSTAINER HYDRAULICS

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	367246.	13089.	36156.
MEAN VALUE	237520.	26177.	72312.
LOWER EXTREME	233195.	38890.	107431.

8 SUSTAINER AIRFRAME

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	4890.	81129.	224112.
MEAN VALUE	1033.	162257.	448224.
LOWER EXTREME	905.	224813.	621032.

9 SUSTAINER PNEUMATICS

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	81550.	21103.	58294.
MEAN VALUE	37294.	42205.	116589.
LOWER EXTREME	35819.	90217.	294219.

10 PROPELLANT UTILIZATION

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	8568.	16985.	46920.
MEAN VALUE	1261.	33970.	93840.
LOWER EXTREME	1017.	48138.	132977.

Figure D-20. Element Cost Vs. Failure Rate
(Sheet 2 of 5)

11 ELECTRICAL

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	175932.	25228.	69690.
MEAN VALUE	105640.	50456.	139380.
LOWER EXTREME	103297.	71843.	198462.

12 GUIDANCE

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	3873.	101661.	280830.
MEAN VALUE	699.	203321.	561660.
LOWER EXTREME	594.	447409.	1235936.

1 FC CABLING

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	8856.	3332.	9204.
MEAN VALUE	2663.	6664.	18409.
LOWER EXTREME	2457.	11544.	31890.

2 FC EXCITATION TRANSF.

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	11820.	57.	156.
MEAN VALUE	4119.	113.	312.
LOWER EXTREME	3862.	199.	550.

3 FC PROGRAMMER

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	46166.	10954.	30259.
MEAN VALUE	26596.	21908.	60518.
LOWER EXTREME	25943.	35549.	98203.

Figure D-20. Element Cost Vs. Failure Rate
(Sheet 3 of 5)

1 AP DISPLACEMENT GYRO

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	121440.	42365.	117031.
MEAN VALUE	86207.	84730.	234062.
LOWER EXTREME	85032.	130262.	359840.

2 AP SERVO AMPL/FILTERS

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	42247.	52264.	144374.
MEAN VALUE	23753.	104527.	288749.
LOWER EXTREME	23137.	227452.	428320.

3 AP RATE GYRO

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	32322.	37350.	103177.
MEAN VALUE	16779.	74700.	206354.
LOWER EXTREME	16260.	126599.	349720.

1 S.S. A - AIRFRAME, RAM

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	1882.	57749.	159528.
MEAN VALUE	367.	115498.	319056.
LOWER EXTREME	266.	205430.	567485.

2 S.S. B - PROPULSION, RAM

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	7617.	345363.	954040.
MEAN VALUE	1604.	690725.	1908080.
LOWER EXTREME	1203.	1272805.	3516037.

Figure D-20. Element Cost Vs. Failure Rate
(Sheet 4 of 5)

3 S.S. C - ELECTRICAL, RAM

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	5916.	11889.	32844.
MEAN VALUE	635.	23779.	65688.
LOWER EXTREME	283.	56782.	156855.

4 S.S. D - GUID/CONTL, RAM

VALUE	FAILURE RATE	HARDWARE COST	ELEMENT COST
UPPER EXTREME	68022.	151167.	417588.
MEAN VALUE	44131.	302334.	835176.
LOWER EXTREME	42539.	489175.	1351313.

Figure D-20. Element Cost Vs. Failure Rate
(Sheet 5 of 5)

BIBLIOGRAPHY

1. Atlas Space Booster Familiarization Course (U); General Dynamics/Astronautics, (No document number); 1 December 1962; Unclassified.

2. Space Flight Operations Plan, Ranger 6 (U); Jet Propulsion Laboratory EPD-78 (Rev. 2); dated 9 December 1963; Unclassified.

3. Standard Space Launch Vehicle Reliability Improvement Program Progress Report (U); General Dynamics/Astronautics BHC 63-001-3; dated 1 September 1963; Unclassified.

4. Reliability Improvement Program Plan for Standard Space Launch Vehicle (SLV-3); General Dynamics GDA 63-0436; dated 20 June 1963; Unclassified.

5. Standard Space Launch Vehicle Reliability Improvement Program Progress Report (U); General Dynamics BHC 63-001-6; dated 1 December 1963; Unclassified.

6. Space Systems Vehicle Test Plan, NASA Program, Model 10205 - Vehicle 6008 (U); Lockheed Missiles and Space Company 1360726; dated 19 June 1963; Unclassified.

7. Sequence of Events, Model 10205, Vehicle 6008 (U); Lockheed Missiles and Space Company 1342657 F; dated 20 September 1962; Unclassified.

8. Basic Equipment List, Ranger 10205-6008, (U); Lockheed Missiles and Space Company, Dept. 91-38; dated 8 October 1963; Unclassified.

9. Analytical Estimation of the Probability of Successful Operation of the Atlas-0004 Autopilot Groups During A Space Booster Mission (U); General Dynamics RSA/TM/27; dated 28 June 1962; Unclassified.

APPENDIX E

REGRESSION

(SCORE Program - Systematic Construction of an Optimum Regression Equation)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (cont)

1 INTRODUCTION

SCORE, an acronym for the expression: "Systematic Construction of an Optimum Regression Equation", is a program developed by LSI for use on this and similar projects. The concepts around which this program has been constructed are discussed in Section 2 following.

For use in systems analysis, information concerning the parameter relationships of an existing physical phenomenon is often presented in the form of bulk data (e.g., data obtained through use of a Monte Carlo approach to system analysis). This fact makes it difficult and often impractical to use such information effectively. The problem is alleviated, however, through the judicious use of regression techniques.

A regression analysis, beginning with given, consistent data on a set of parameters, proceeds to generate an analytical expression describing the behavior of one parameter, the dependent variable, in terms of the remaining parameters, the independent variables, and their interactions.

The regression analysis program is based on the Stepwise Regression with Simple Learning digital computer program developed at the University of Michigan. This is a unique approach to multiple regression curve fitting of data in that the terms of the equation are generated by the program.

2 REGRESSION TECHNIQUES

2.1 LINEAR REGRESSION

Data consisting of two variables is discussed first since the curve fitting approach may be illustrated easily by means of a plot. Then follows a discussion of data consisting of many variables wherein the process will be shown to be an extension of that used for two variables.

2.1.1 Two Dimensional

If a set of data consisting of corresponding values of two variables, say x and y, is given, the problem is to find an approximating curve which describes the relationship between the two.

2.1.1.1 Straight Line

The simplest type of approximating curve is a straight line, whose equation can be written:

$$y = a_0 + a_1 x . \qquad \text{Eq. 1}$$

For the case where only two data points are involved, this expression will give an exact fit. However, since many data points are generally given, a straight line relationship is not likely to provide an exact fit. It then becomes a matter of finding the "best" straight line passing through the data points.

In order to establish a criterion for a "best" fit, consider the plot of x vs. y in Figure E-1. For a given value of x, say x_1, there will be a difference between the value y_1 and the corresponding value as determined from the straight line L. As noted in Figure E-1, this difference is denoted by d_1, which is often referred to as a deviation, error, or residual and which may be positive, negative,

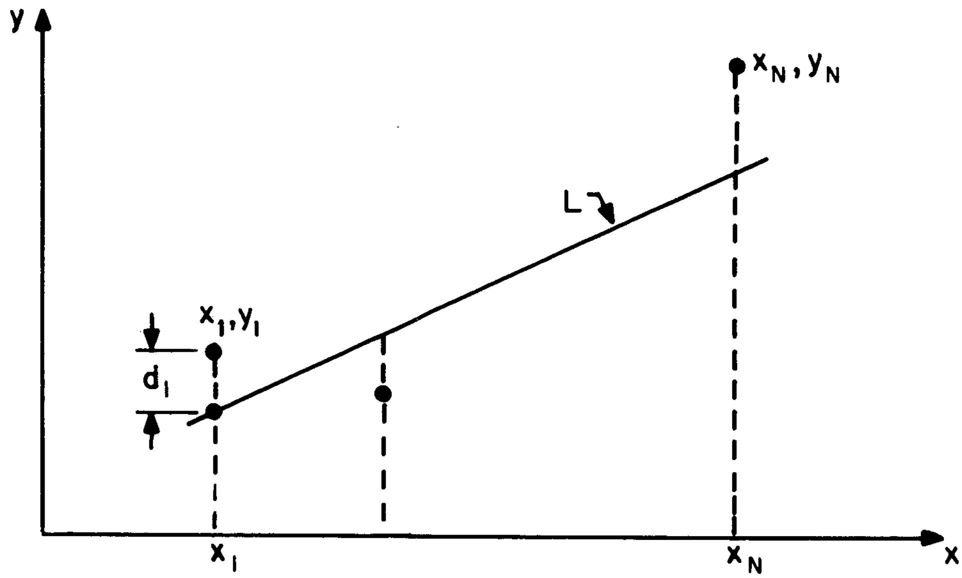


Figure E-1. Linear Approximation for a Set of Data Points

or zero. A measure of the "goodness of fit" of the line L to the given data is provided by the quantity

$$\Delta = d_1^2 + d_2^2 + \ldots + d_N^2 \qquad \text{Eq. 2}$$

where N is the number of points in the data set.

If this quantity is small, the fit is good, if it is large, the fit is poor. The line which minimizes this quantity for a given data set will be called the best straight line in the "least squares" sense or the least squares line.

The method used to compute a_1 and a_0 for this least squares line is developed as follows. Expressing Equation 2 in terms of the line itself and the data points gives:

$$\Delta = (y_1 - a_0 - a_1 x_1)^2 + (y_2 - a_0 - a_1 x_2)^2 + \ldots + (y_N - a_0 - a_1 x_N)^2 . \quad \text{Eq. 3}$$

It is desired to find values for a_1 and a_0 such that Δ is a minimum. Therefore, the following equations must be satisfied:

$$\frac{\partial \Delta}{\partial a_1} = 0 , \quad \text{Eq. 4}$$

$$\frac{\partial \Delta}{\partial a_0} = 0 . \quad \text{Eq. 5}$$

Investigation of the second partials shows these conditions to be sufficient for a minimum.

Substituting Equation 3 into Equations 4 and 5 gives:

$$2 (y_1 - a_0 - a_1 x_1)(-x_1) + 2 (y_2 - a_0 - a_1 x_2)(-x_2) + \ldots + 2 (y_N - a_0 - a_1 x_N)(-x_N) = 0 . \quad \text{Eq. 6}$$

$$2 (y_1 - a_0 - a_1 x_1)(-1) + 2 (y_2 - a_0 - a_1 x_2)(-1) + \ldots + 2 (y_N - a_0 - a_1 x_N)(-1) = 0 . \quad \text{Eq. 7}$$

Collecting coefficients of a_1 and a_0 gives, for Equation 6:

$$a_1 (2 x_1^2 + 2 x_2^2 + \ldots + 2 x_N^2) +$$

$$a_0 (2 x_1 + 2 x_2 + \ldots + 2 x_N) + \qquad \text{Eq. 8}$$

$$(- 2 x_1 y_1 - 2 x_2 y_2 - \ldots - 2 x_N y_N) = 0 .$$

and for Equation 7:

$$a_1 (2 x_1 + 2 x_2 + \ldots + 2 x_N) +$$

$$a_0 (2 + 2 + \ldots + 2) + \qquad \text{Eq. 9}$$

$$(- 2 y_1 - 2 y_2 - \ldots - 2 y_N) = 0 .$$

Simplifying and rearranging into matrix form yields

$$\begin{bmatrix} N & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 \end{bmatrix} \text{x} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i y_i \end{bmatrix} . \qquad \text{Eq. 10}$$

The solution of these two simultaneous equations for a_0 and a_1 gives the coefficients for the least squares line.

2.1.1.2 Polynomial

Data which does not exhibit a linear relationship may often be "fitted" very well using a polynomial of the form:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1} + a_n x^n . \qquad \text{Eq. 11}$$

The degree n of the polynomial being fitted is a function of the accuracy required in the fit. As the degree of the polynomial is increased, the goodness of fit increases. The least squares coefficients are computed by solving the following simultaneous equations for the a's:

$$\begin{bmatrix} N & \Sigma x & \Sigma x^2 \ldots & \Sigma x^{n-1} & \Sigma x^n \\ \Sigma x & \Sigma x^2 & \Sigma x^3 \ldots & \Sigma x^n & \Sigma x^{n+1} \\ \Sigma x^2 & \Sigma x^3 & \Sigma x^4 \ldots & \Sigma x^{n+1} & \Sigma x^{n+2} \\ \vdots & \vdots & & \vdots & \vdots \\ \Sigma x^{n-1} & \Sigma^{n} & \ldots\ldots & \Sigma x^{2n-2} & \Sigma x^{2n-1} \\ \Sigma x^n & \Sigma x^{n+1} & \ldots\ldots & \Sigma x^{2n-1} & \Sigma x^{2n} \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix} = \begin{bmatrix} \Sigma y \\ \Sigma xy \\ \Sigma x^2 y \\ \vdots \\ \Sigma x^{n-1} y \\ \Sigma x^n y \end{bmatrix} .$$

Eq. 12

Note the shortened notation Σx, Σxy, etc., in place of $\sum_{i=1}^{N} x_i$, $\sum_{i=1}^{N} x_i y_i$, etc. This notation will be used throughout the report to eliminate confusion.

2.1.1.3 Functional

Theoretically, an nth degree polynomial can be made to fit any set of two-dimensional data simply by making n large enough. This may, however, result in a very large equation. Consideration of other forms of the independent variable in the equation often reduces the size of the equation required for a good fit. Such forms might be:

$x^{1/3}$, $x^{-1/2}$, $\sin x$, e^x

and similar types.

Assume an equation of the form:

$$y = a_0 + a_1 e^x + a_2 \sin x + a_3 x^{-2} + a_4 x^{1/3} . \qquad \text{Eq. 13}$$

For this case, the least squares coefficients are computed by solving the following simultaneous equations for the a's:

$$\begin{bmatrix} N & \Sigma e^x & \Sigma \sin x & \Sigma x^{-2} & \Sigma x^{1/3} \\ \Sigma e^x & \Sigma e^{2x} & \Sigma e^x \sin x & \Sigma e^x x^{-2} & \Sigma e^x x^{1/3} \\ \Sigma \sin x & \Sigma e^x \sin x & \Sigma \sin^2 x & \Sigma x^{-2} \sin x & \Sigma x^{1/3} \sin x \\ \Sigma x^{-2} & \Sigma e^x x^{-2} & \Sigma x^{-2} \sin x & \Sigma x^{-4} & \Sigma x^{1/3} x^{-2} \\ \Sigma x^{1/3} & \Sigma e^x x^{1/3} & \Sigma x^{1/3} \sin x & \Sigma x^{1/3} x^{-2} & \Sigma x^{2/3} \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} \Sigma y \\ \Sigma e^x y \\ \Sigma y \sin x \\ \Sigma x^{-2} y \\ \Sigma x^{1/3} y \end{bmatrix}$$

Eq. 14

At this point, it is pointed out, the normal equations can be easily written for any equation form which is linear in the "a" coefficients. Observe that the first equation in Equation Set 14 is formed by summing on both sides of Equation 13, i.e.:

$$\Sigma y = a_0 N + a_1 \Sigma e^x + a_2 \Sigma \sin x + a_3 \Sigma x^{-2} + a_4 \Sigma x^{1/3} . \quad \text{Eq. 15}$$

The second equation is formed by first multiplying both sides of Equation 13 by the variable whose coefficient is a_1, namely e^x, and then summing. The third equation is formed similarly, this time multiplying first by sin x. Note that this is not a derivation of the equations but simply a means for remembering them.

Thus far only problems which are two dimensional in nature have been considered. The approach to multi-dimensional problems is an extension of the approach to the two dimensional case.

2.1.2 Multi-Dimensional

Data consisting of more than two variables also lends itself to least squares curve fitting, through extension of the concepts discussed above.

2.1.2.1 Linear in the Independent Variables

Assume a set of data consisting of N consistent sets of n variables. The linear equation relating these variables would be:

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + \ldots + a_n x_n . \quad \text{Eq. 16}$$

The simultaneous equations which yield the least squares values for the "a" coefficients are as follows:

$$\begin{bmatrix} N & \Sigma x_1 & \Sigma x_2 & \cdots & \Sigma x_n \\ \Sigma x_1 & \Sigma x_1^2 & \Sigma x_1 x_2 & \cdots & \Sigma x_1 x_n \\ \Sigma x_2 & \Sigma x_1 x_2 & \Sigma x_2^2 & \cdots & \Sigma x_2 x_n \\ \Sigma x_3 & \Sigma x_1 x_3 & \Sigma x_2 x_3 & \cdots & \Sigma x_3 x_n \\ \vdots & \vdots & \vdots & & \vdots \\ \Sigma x_n & \Sigma x_1 x_n & \Sigma x_2 x_n & \cdots & \Sigma x_n^2 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \Sigma y \\ \Sigma x_1 y \\ \Sigma x_2 y \\ \Sigma x_3 y \\ \vdots \\ \Sigma x_n y \end{bmatrix} . \quad \text{Eq. 17}$$

Solving these equations for the a's gives the least squares coefficients for Equation 16.

2.1.2.2 Cross Products and Functions of Independent Variables

Further generalization of the application of least squares curve fitting for the multi-dimensional problem is possible. Assume that the form of the equation relating the variables in a five-dimensional set of data is as follows:

$$y = a_0 + a_1 x_1^{1/3} x_3^2 + a_2 x_2^{-3} x_4^{1/4} + a_3 x_3 \cos x_1 . \quad \text{Eq. 18}$$

At this point, the term T, in general, is defined as a product of functions of the independent variables under consideration. Referring to Equation 18 the following terms are thus defined:

$$T_1 = x_1^{1/3} x_3^2$$

$$T_2 = x_2^{-3} x_4^{1/4}$$

$$T_3 = x_3 \cos x_1 .$$

The terms T_1, T_2, and T_3 may now be thought of as the independent variables while retaining y as the dependent variable. Equation 18 then may be written as:

$$y = a_0 + a_1 T_1 + a_2 T_2 + a_3 T_3 \qquad \text{Eq. 19}$$

and the least squares fitting procedure as discussed in Section 2.1.2.1 may be carried out.

Thus far the flexibility of least squares curve fitting has been demonstrated showing that a variety of problem types can be handled provided that the equation form is known a priori. However, the practical situation is often such that there exists very little information regarding the form of the equation. The next logical step, then, towards effectively using the computer to establish data relationships is to have the computer generate the form of the equation as well as the coefficients. This is accomplished by systematically constructing the equation, term by term, until a suitable equation is obtained. Due to the stepping process involved, this procedure is commonly referred to as Stepwise Regression.

2.1.3 Stepwise Regression

A basic approach towards the computerized construction of the form of a regression equation as well as its coefficients is discussed in Reference 1. The equation desired is assumed to be linear in the variables as well as in the coefficients. Thus, for data consisting of N sets of n independent variables, x_1, x_2, ..., x_n, and one dependent variable, y, the form of the equation including all of the variables would be:

$$y = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_n x_n . \qquad \text{Eq. 20}$$

In general, the contribution of each one of the variables x_1 through x_n towards explaining the dependent variable y varies considerably. In fact some of the independent variables may not be related to y at all. It is desirable, then, to have only those variables in the equation which are truly significant. This is accomplished by inserting variables one at a time, each time selecting for insertion that variable which would make the greatest contribution to the (as yet,) unexplained portion of the dependent variable. Now follows a discussion of the manner in which these contributions are calculated.

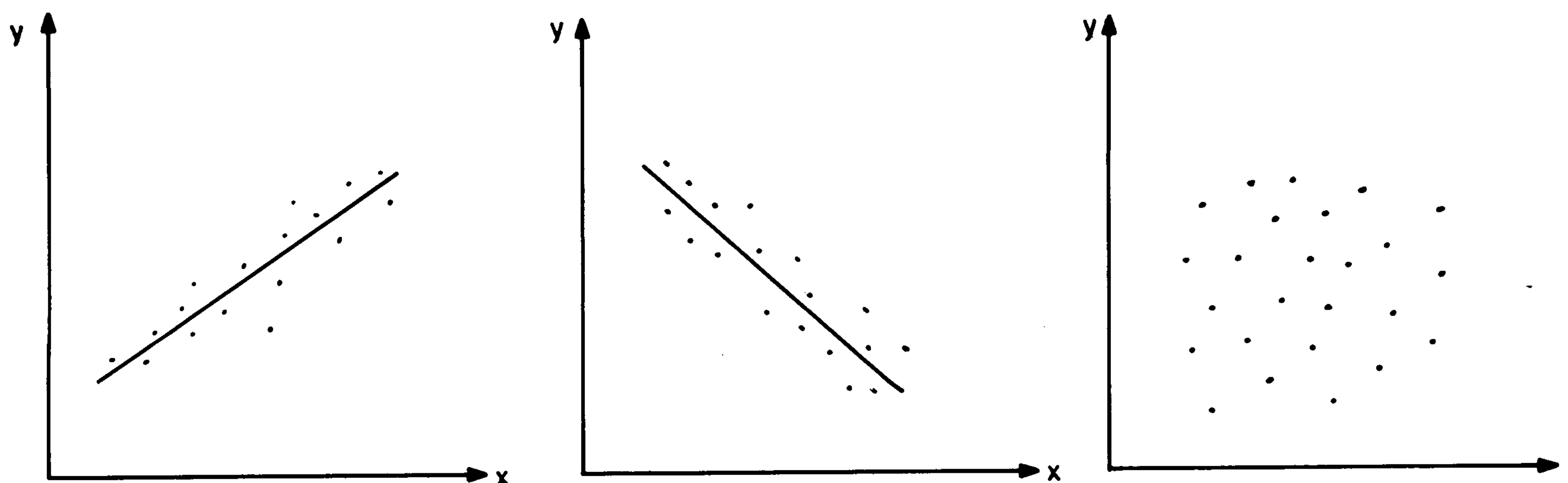
Prior to insertion of any of the variables into the regression equation, the linear correlation coefficients relating each of the independent variables to the dependent variable are calculated. The linear correlation coefficient for the ith variable, x_i, paired with the dependent variable y is given by:

$$r(x_i, y) = \frac{\frac{\Sigma x_i y}{N} - \frac{\Sigma x_i \Sigma y}{N^2}}{\sigma_{x_i} \sigma_y} \qquad \text{Eq. 21}$$

where σ denotes the standard deviation of the variable appearing as its subscript. The linear correlation coefficient is a measure of how well a straight line explains the relationship between the two variables, x_i and y. Scatter diagrams showing the location of points (x_i, y) on a rectangular coordinate system for various types of correlation are shown in Figure E-2. If all points in this scatter diagram seem to lie near a line, as in (a) and (b) of Figure E-2, the correlation is called linear and the magnitude of r as given by Equation 21 will approach unity. If y tends to increase as x_i increases, as in Figure E-2a, the correlation is called positive or direct correlation and r will assume a positive value. If y tends to decrease as x_i increases, as in Figure E-2b, the correlation is called negative or inverse correlation, and r will assume a negative value. If there is no relationship indicated between the variables as in Figure E-2c, there is no linear correlation between them and r will assume a value of zero.

The first step in constructing a regression equation consists of finding the best single predictor of y from among the x variables. This is done by scanning the list of linear correlation coefficients between each variable in turn and y. That variable having the greatest (in absolute value) correlation with y is selected to form the first predicting equation. Should this variable be x_3, the first predicting equation would be:

$$y = a_0 + a_3 x_3. \qquad \text{Eq. 22}$$



a.POSITIVE LINEAR CORRELATION b.NEGATIVE LINEAR CORRELATION c. NO CORRELATION

Figure E-2. Scatter Diagrams for Various Types of Correlation

The a_0 and a_3 values are computed using the least squares criterion as discussed previously.

Once a predicting equation is formed, subsequent steps are of a slightly different form. First, an importance factor, or F-level, is calculated for each variable in the equation. These F-levels measure the relative importance of each variable in the equation to the prediction of the dependent variable. The smallest of these F-levels is then isolated and if it is less than some specified critical F-level for removing variables, that variable is removed from the equation. This process is desirable since the contribution of a particular variable may become insignificant if a combination of other variables in the equation explains the dependent variable better than the single variable. The F-level for removing the variable x_i , $F_r(x_i)$, is computed as:

$$F_r\ (x_i) = \left[\frac{\Sigma(y - y_e)^2 - \Sigma(y - y'_e)^2}{\Sigma(y - y_e)^2} \right] x\, \phi \qquad \text{Eq. 23}$$

where

y_e = the estimated value of y using the regression equation excluding the variable x_i

y'_e = the estimated value of y using the regression equation including the variable x_i

ϕ = the number of degrees of freedom, which equals the number of data sets minus the number of variables in the regression equation minus one.

If all the F-levels for removing variables in the equation are greater than the specified critical value, the next step is to select a variable not in the equation for possible insertion into the equation. For each of these variables, a potential importance factor or F-level for entering is computed. These F-levels measure the relative contribution each variable not in the equation would make towards explaining the dependent variable if it were put into the equation. The largest of these is isolated, thereby selecting the best variable for entering into the equation at that point. The corresponding F-level is then compared with a specified critical F-level for entering variables. If the F-level exceeds this value, the variable is inserted into the equation. The F-level for entering a variable x_i , $F_e(x_i)$, is computed as:

$$F_e(x_i) = \left[\frac{\Sigma(y - y_e)^2 - \Sigma(y - y_e')^2}{\Sigma(y - y_e')^2} \right] x\ \phi. \qquad \text{Eq. 24}$$

This process of inserting and removing variables is continued until there is no variable not in the equation whose F-level is greater than the level for entering, and also there is no variable in the equation whose F-level is less than the level for removing.

In order to clarify the above procedure, an example is used. Assume a data set consisting of five independent variables x_1 , x_2, x_3 , x_4 , and x_5 and a dependent variable y.

The first step consists of finding the best single predictor of y from the five independent variables. For the first step, this corresponds to picking that variable having the greatest (in absolute value) linear correlation coefficient when paired with y. This variable will also be the variable having the highest F-level for entering. If this F-level value exceeds the specified critical level, the variable will be entered into the equation. If this variable were x_3 , the first predicting equation would be:

$$y = a_0 + a_3 x_3 \qquad \text{Eq. 25}$$

where the values for a_0 and a_3 are computed using the method of least squares.

The F-level for entering variable x_3 is computed according to Equation 24 as:

$$F_e(x_3) = \left[\frac{\Sigma(y - \bar{y})^2 - \Sigma(y - a_0 - a_3 x_3)^2}{\Sigma(y - a_0 - a_3 x_3)^2} \right] x\ \phi. \qquad \text{Eq. 26}$$

Note that $\bar{y}$ or the mean value of the dependent variable is substituted for y_e as given in Equation 24. This is done for the first step only, since prior to entry of any of the independent variables into the equation, $\bar{y}$ is the best estimate of the dependent variable. Also, for the first step, ϕ is equal to N-2, where N is the number of data sets.

The next step involves computation of F-levels for entering for each of the remaining variables not in the equation, i.e.; x_1, x_2, x_4, and x_5. That variable having the highest value for F_e is then isolated and its F_e value compared to the critical F-level for entry as prescribed by the user. Assume this variable to be x_4. The regression equation then becomes:

$$y = a'_0 + a'_3 x_3 + a_4 x_4 \quad . \qquad \text{Eq. 27}$$

Note the primes on a_0 and a_3. These coefficients no longer have the same value after adding x_4 as they had in Equation 25. As before, these coefficients are all computed using the method of least squares.

The F-level for entering variable x_4 is computed according to Equation 24 as:

$$F_e\,(x_4) = \left[\frac{\Sigma(y - a_0 - a_3 x_3)^2 - \Sigma(y - a'_0 - a'_3 x_3 - a_4 x_4)^2}{\Sigma(y - a'_0 - a'_3 x_3 - a_4 x_4)^2} \right] x\,\phi \quad . \qquad \text{Eq. 28}$$

In this case, ϕ is equal to N-3 since the number of degrees of freedom decreases by 1 for each succeeding step in the equation building process.

At this point an F-level for removing variables is computed for each variable in the equation. This is desirable since the contribution of a particular variable may decrease considerably as other variables are inserted into the equation.

Suppose that for the third step no variables were removed and variable x_1 was added. The regression equation is then:

$$y = a''_0 + a_1 x_1 + a''_3 x_3 + a'_4 x_4 . \qquad \text{Eq. 29}$$

Assume that the combination of x_1 and x_4 makes the contribution of x_3 negligible. This will be apparent by examining the F-level for removing variables as given by Equation 23. The estimated value for y using the current equation excluding x_3 would be:

$$y_e = a'''_0 + a'_1 x_1 + a''_4 x_4 . \qquad \text{Eq. 30}$$

Equation 23 then becomes

$$F_r (x_3) = \left[\frac{\Sigma(y - a'''_0 - a'_1 x_1 - a''_4 x_4^2) - \Sigma(y - a''_0 - a_1 x_1 - a''_3 x_3 - a'_4 x_4^2)}{\Sigma(y - a''_0 - a_1 x_1 - a''_3 x_3 - a'_4 x_4^2)} \right] x \phi .$$

Eq. 31

A value for $F_r(x_3)$ less than the critical value as prescribed by the user indicates that the contribution of x_3 is less than that required in order to retain a variable in the regression equation. Equation 30 then becomes the current regression equation.

The next step may indicate that x_2 should be in the regression equation. The equation then becomes:

$$y = a''''_0 + a''_1 x_1 + a_2 x_2 + a'''_4 x_4 . \qquad \text{Eq. 32}$$

If, on the following step, it is concluded that no variables can be removed or inserted, the final equation would be as shown by Equation 32. A more rigorous and technical discussion of this process may be found in Reference 3.

3 THE SCORE PROGRAM

The word "SCORE" is an acronym for the expression "Systematic Construction of an Optimum Regression Equation". The SCORE program herein described has been constructed around the concepts already discussed in Section 2.

3.1 PHILOSOPHY

Basically, the approach used is that discussed in Section 2.1.3 with the added flexibility of considering terms which are cross products and functions of the independent variables rather than simply the independent variables themselves. Since the number of possible terms which can be constructed in this manner soon approaches infinity, the bulk of the problem is involved in constructing and choosing terms and "learning" as various terms are accepted or rejected for use in the regression equation.

3.1.1 Term Characteristics

Typical terms in a regression equation involving the four independent variables x_1 , x_2 , x_3, and x_4 would be:

$$x_1^2 \, x_2^{-3} \, ,$$

$$x_2 \, x_3 \, x_4 \, ,$$

and

$$x_3^{-1} \, .$$

Terms such as these may be defined by three basic characteristics:

- Interaction Order
- Variables Involved
- Functions of Variables.

The interaction order of a term is defined to be the number of distinct independent variables involved in the term. Thus a term such as $x_1^2 \; x_3^{1/3}$ has an interaction order of two, since two independent variables (x_1 and x_3) are involved.

The variables involved in a term are those independent variables which make up the term. Thus, for the above example $x_1^2 \; x_3^{1/3}$, the independent variables are x_1 and x_3 . The number of variables which must be defined for a particular term is equal to the interaction order specified for that term.

The functions of the variables involved define the power or exponent assigned to each variable in the term. Thus, the functions for the term $x_1^2 \; x_3^{1/3}$ are x^2 and $x^{1/3}$ for variables x_1 and x_3 respectively. Thus, the number of functions which must be defined for a particular term is also equal to the interaction order specified for that term.

Term construction involves a weighted random selection of each of these characteristics.

3.1.2 Weighted Random Selection

Implementation of a random selection procedure on the digital computer is facilitated through use of a random number generator. The procedure used for generating random numbers between 0 and 1 on the present IBM 1620 computer involves use of a modified Fibonacci sequence. Details on this approach and its implementation on the computer may be found in the LSI Technical Note AAT-TN-64-001. *

Random selection of one item from a collection of n discrete items using a random number generator is illustrated in Figure E-3.

Dividing the abscissa into n equally spaced intervals and using a linear ordinate scale between 0 and 1, a one-to-one correspondence may be established between any random number from 0 to 1 and a specific item number. A straight line OA constructed through the origin (O) and the point on the upper boundary of both axes (A) establishes this relationship. Thus, for a random number RN_1 , entering the graph shown in Figure E-3 with the value of RN_1 on the ordinate axis and going straight across parallel to the abscissa axis gives an intersection of OA in the Item Number 5 region. Thus RN_1 indicates Item Number 5 be selected from the n possible candidates. Similarly, a random number equal to RN_2 selects Item Number n, and RN_3 selects Item Number 3.

* AAT-TN-64-001 -- "A Random Number Generator Based on Truncated Fibonacci Numbers": Morin, G. Rex; Dec. 64

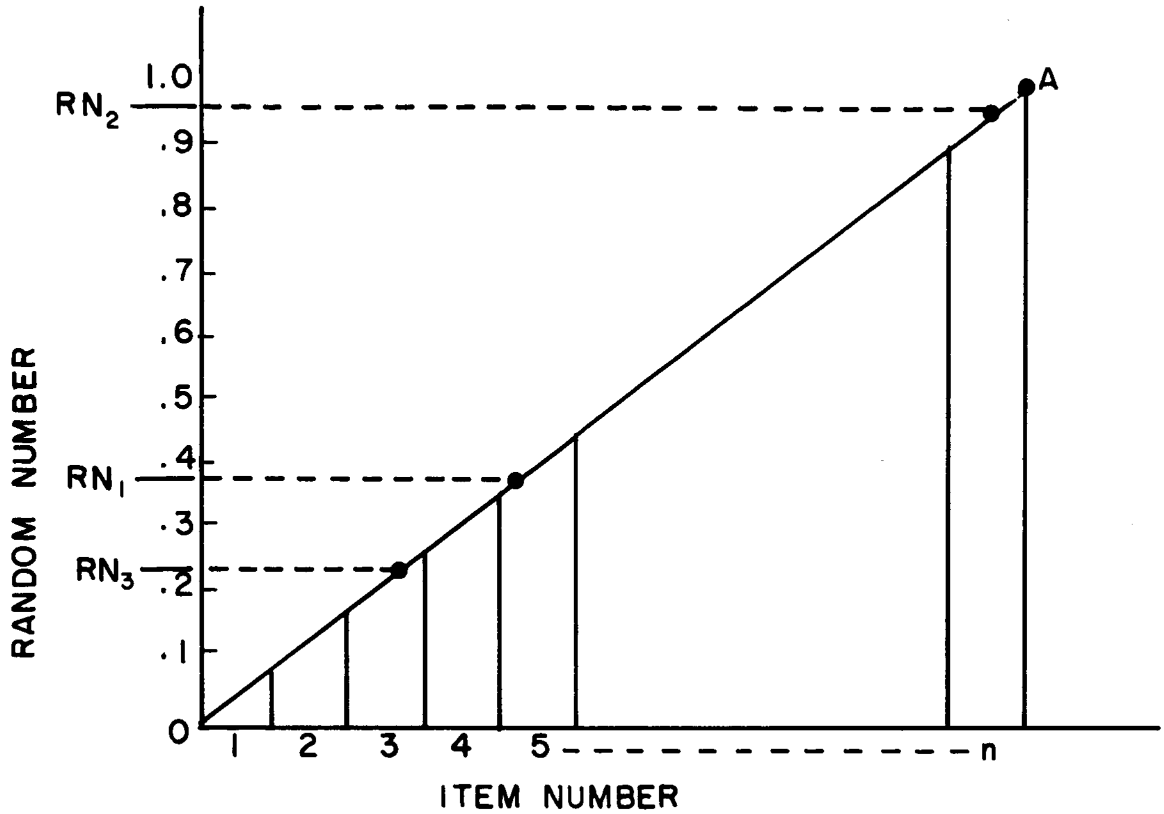


Figure E-3. Random Selection of Items From a Collection of n Discrete Items Using a Random Number Generator

Using this process and assuming a number generator which is truly random between 0 and 1, a truly random selection of items is obtained.

It is often desirable to weight the selection process so that some items are more apt to be selected than others. This may be implemented in a manner very similar to that discussed above. Figure E-3 may be thought of as the integral of an item weighting function as shown in Figure E-4 which has been normalized to give a total area of 1.

Adjusting these weights so that each weight is proportional to the desired probability of selection might yield a weighting function as shown in Figure E-5.

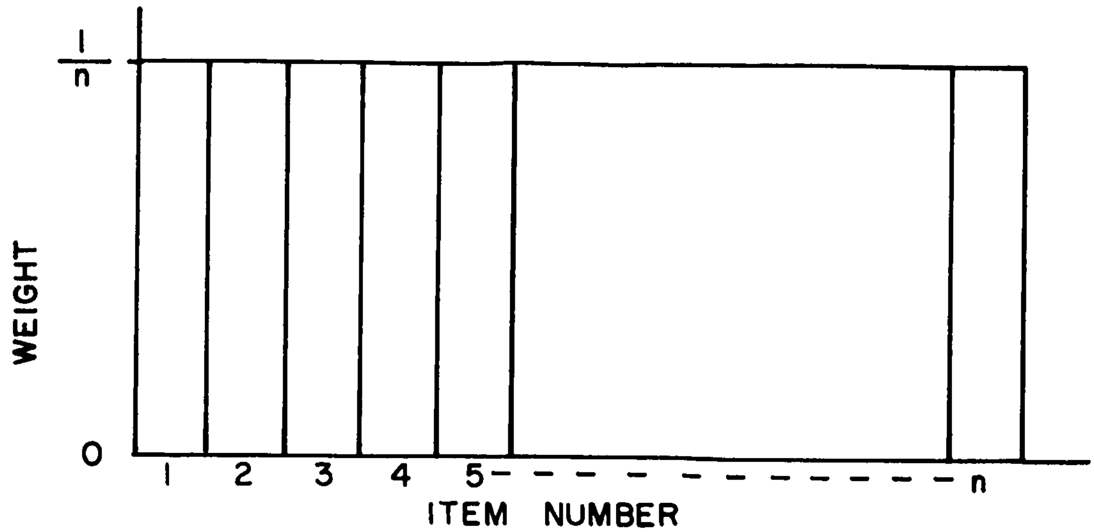


Figure E-4. Item Selection Weights Such That the Selection of All Items Is Equally Likely

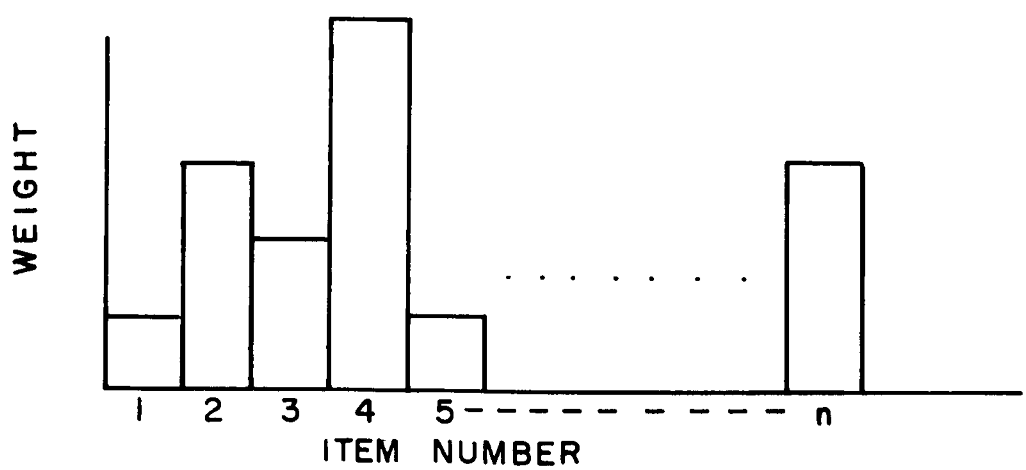


Figure E-5. Item Selection Weights Such That the Selection of All Items Is Not Equally Likely

Integrating this function gives the plot shown in Figure E-6.

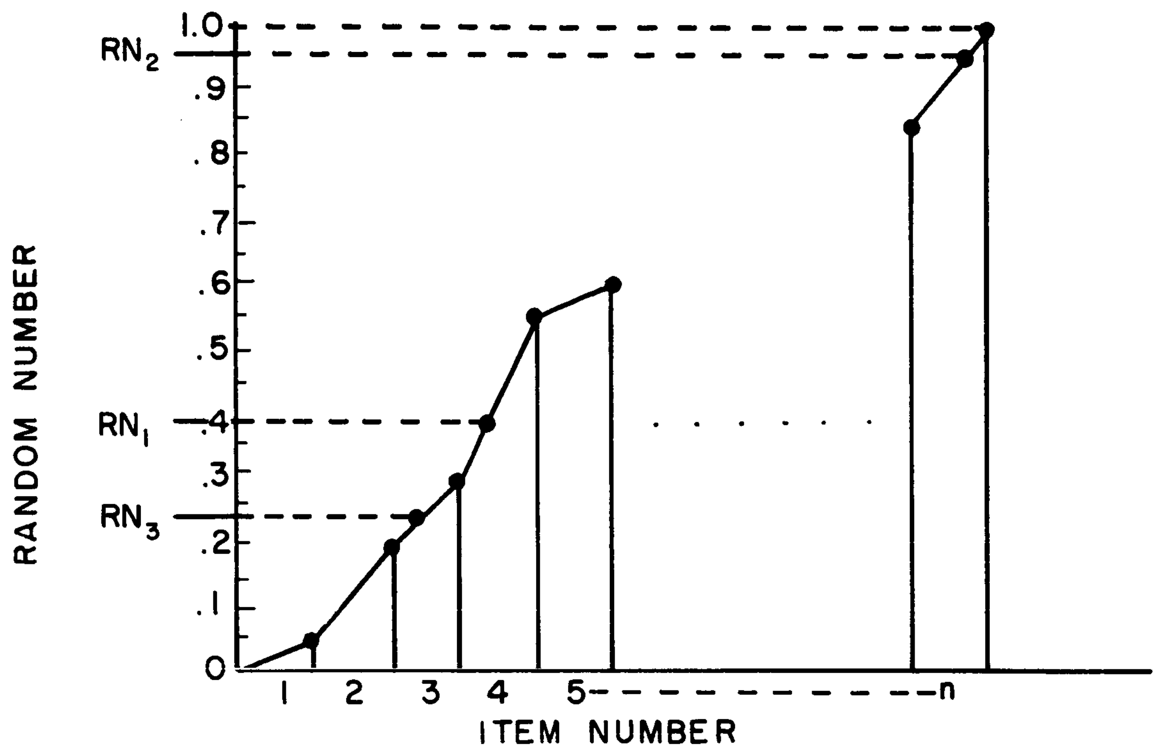


Figure E-6. Weighted Random Selection of Items From a Collection of n Discrete Items Using a Random Number Generator

Entering Figure E-6 with a random number as described above will result in a weighted random selected of the candidate items.

3.1.3 Term Construction

The first step in constructing a term is the weighted random selection of an interaction order. One of the inputs to the SCORE program is the maximum interaction order to be considered in the construction of terms. Selection weights to be assigned to each interaction order may also be entered as inputs, if desired. If no weights are entered, each interaction order will be assigned the same weighting value initially. A process simulating the use of a graph such as shown in Figure E-6 is then carried out by the computer.

Once an interaction order for a term has been established, the variables to be used in that term are selected on a weighted random basis. The number of variables which must be selected is equal to the interaction order selected for the term. Selection weights for each of the independent variables may be entered as inputs. If they are not entered they will initially be considered equal for each variable. After selection of a variable for a particular term, the selection weight for that variable is set to zero temporarily to insure against selecting that same variable again for the same term. After all the variables for a term are selected, the selection weights are restored to the values existing just prior to term construction.

The final step involves assigning a function to each of the variables involved in the term being constructed. A set of function selection weights is maintained for each of the independent variables. The number of functions to be considered for each independent variable is an input to the SCORE program. The functions currently available are shown in Table E-I.

Specification of six functions to be considered means function Numbers 1 through 6 as shown in Table E-I would be considered. Function selection weights may also be specified as inputs to the SCORE program. If no weights are specified all of the functions to be considered are initially weighted equally.

Additional functions such as cos x, e^x and others will be made available in the SCORE program in the future.

In summary, the initial step in the SCORE program is construction of a set of terms from which a regression equation will be constructed according to the procedure described in Section 2.1.3, Stepwise Regression. The independent variables designated by x's in that section are now replaced by terms as described in this section. The number of terms from which an equation is constructed is specified as an input variable to the program. Due to the random nature of selecting terms, the possibility exists that two or more terms might be identical in a given set. Since this would be a waste of computer time, a check is made during term construction to insure against this occurrence.

Once a regression equation has been constructed, the terms which did not become part of the equation are abandoned and new terms are constructed to replace them. The equation building process then starts over again using the terms which were successful in the previous equation plus the terms which were constructed to replace the previously abandoned terms. The old equation is not retained as such. This process continues until a previously established accuracy requirement is met or until manual intervention is made.

Table E-I. Variable Functions Currently Available for Use in the SCORE Program

Function Number	Function
1	x
2	x^{-1}
3	x^{2}
4	x^{-2}
5	$x^{1/2}$
6	$x^{-1/2}$
7	x^{3}
8	x^{-3}
9	$x^{1/3}$
10	$x^{-1/3}$
⋮	⋮
19	x^{6}
20	x^{-6}

3.1.4 Learning

Each time a new regression equation has been constructed from a given set of terms, the learning process may be implemented. Of the set of terms available, a certain number have successfully entered the equation and the remainder of them were not successful. It would then be desirable to direct the future selection process toward those terms which are similar to the terms which were successful and away from those similar to the unsuccessful terms. This is done by adjusting the selection weights associated with the successful and unsuccessful terms.

Each time a particular interaction order is successful, the corresponding selection weight is increased. The present SCORE program increases the weight by 25%. Each time a particular interaction order is not successful, the corresponding selection weight is decreased. The current decrease is 20%. The same procedure is followed for the independent variables themselves. Each time a particular variable is used, its corresponding selection weight is increased by 25% and each time it is rejected, its selection weight is decreased by 20%. Finally, this adjustment is applied to the set of function selection weights existing for each variable.

After these adjustments have been made, the weights are renormalized so as to give a total of 1.0 for their sum.

3.1.5 Equation Characteristics

After construction of an equation, the standard error, the coefficient of determination, and the multiple correlation coefficient for the dependent variable are computed.

The standard error of y is given by:

$$s_y = \sqrt{\frac{\Sigma (y - y_e)^2}{\phi}} \qquad \text{Eq. 33}$$

where

y = the dependent variable in the set of data

y_e = the estimated value of the dependent variable using the regression equation

ϕ = the number of degrees of freedom which equals the number of data sets minus the number of variables in the regression equation minus one.

The coefficient of determination is given by:

$$r_y^2 = 1. - \frac{\Sigma(y - y_e)^2}{\Sigma(y - \bar{y})^2} \quad . \qquad \text{Eq. 34}$$

The coefficient of multiple correlation is simply the square root of the coefficient of determination, or:

$$r_y = \pm \sqrt{1. - \frac{\Sigma(y - y_e)^2}{\Sigma(y - \bar{y})^2}} \quad . \qquad \text{Eq. 35}$$

A discussion of the significance of these parameters is given in Reference 2.

The least squares coefficients of the terms in the regression equation are also computed along with the standard error of each of these coefficients. The standard error of a coefficient is a measure of the statistical error existing in the coefficient. If the interval described by the coefficient value plus and minus its standard error is formed, this interval can be expected to include the correct value of the coefficient in about 68% of all cases.

The program continues to construct equations until an equation is found which meets the previously specified criterion for accuracy of fit or until manual intervention is made. The criterion currently used for accuracy is the magnitude of the standard error of the dependent variable as given by Equation 33.

3.1.6 Selection of Critical F-Levels for Entering and Removing Variables

Previously mentioned in this report were the critical values of F-level for entering or for removing a variable of the equation. Both of these critical values are inputs to the SCORE program. Therefore, they will be discussed here in greater detail.

Consider first the F-level for entering and the specified critical value for the F-level. These are used in making the decision of whether or not to enter a variable into the regression equation. The computed F-level for entering x_i as given by Equation 24 is proportional to the ratio of the additional amount of variation in the y variable which could be explained or predicted by x_i if it were entered in the regression equation,to the as yet unexplained variation in y. If this ratio is small, there is some question as to whether the apparent potential contribution of that variable is real or if it is due to noise or extraneous information on the data values. Therefore, it is possible that a variable will be entered into the regression equation when it should not really be there. The use of F-levels is an attempt at controlling the likelihood of committing such an error. The method of choosing a critical F-level for entering a variable is as follows:

a. Choose a maximum allowable likelihood of committing the error. Call this p ($p = 0$ corresponds to never in error, $p = 0.05$ corresponds to 1 chance in 20 of committing the error).

b. Compute $\phi = N-K-1$, where N is the number of data sets involved, and K is the number of variables expected in the regression equation (ϕ = degrees of freedom).

c. Using Table E-II, find the entry under the column headed by p and opposite the row labeled n. This entry is the proper critical value of F-level for entering. Interpolation of the values in this table may be used.

Table E-II. F-Levels for Various Error Probabilities and Degrees of Freedom

n	p=.99	p=.95	p=.90	p=.50	p=.10	p=.05	p=.025	p=.01	p=.001
2	.00020	.0050	.020	.667	8.53	18.5	38.5	98.5	998
3	.00019	.0046	.019	.585	5.54	10.1	17.4	34.1	167
4	.00018	.0044	.018	.549	4.54	7.71	12.2	21.2	74.1
5	.00017	.0043	.017	.528	4.06	6.61	10.0	16.3	47.2
6	.00017	.0043	.017	.515	3.78	5.99	8.81	13.7	35.5
7	.00017	.0042	.017	.506	3.59	5.59	8.07	12.2	29.2
8	.00017	.0042	.017	.499	3.46	5.32	7.57	11.3	25.4
9	.00017	.0040	.017	.494	3.36	5.12	7.21	10.6	22.9
10	.00017	.0041	.017	.490	3.28	4.96	6.94	10.0	21.0
11	.00016	.0041	.017	.486	3.23	4.84	6.72	9.65	19.7
12	.00016	.0041	.016	.484	3.18	4.75	6.55	9.33	18.6
15	.00016	.0041	.016	.478	3.07	4.54	6.20	8.68	16.6
20	.00016	.0040	.016	.472	2.97	4.35	5.87	8.10	14.8
24	.00016	.0040	.016	.469	2.93	4.26	5.72	7.82	14.0
30	.00016	.0040	.016	.466	2.88	4.17	5.57	7.56	13.3
40	.00016	.0040	.016	.463	2.84	4.08	5.42	7.31	12.6
60	.00016	.0040	.016	.461	2.79	4.00	5.29	7.08	12.0
120	.00016	.0039	.016	.458	2.75	3.92	5.15	6.85	11.4
∞	.00016	.0039	.016	.455	2.71	3.84	5.02	6.63	10.8

It should be noted here that as the number of data sets becomes large, (i.e. > 20) these values remain essentially constant for a given p. Typical values for p for entering are 0.05 to 0.10. Therefore, for N > 20, values of the critical F-level for entering will range between 2 and 5. It is further possible to commit the error of retaining a variable in the regression equation when it should have been removed. The same discussion as given above applies here. However, the restriction imposed is that the critical F-level for entering variables be at least as great as the F-level for removing variables. Therefore, for N > 20, values of the critical F-level for removing will usually range between 1 and 4.

3.2 OPERATION

A step by step description of how the SCORE program may be used, along with Input Data Transmittal forms for data preparation, will now be given.

The program is designed to handle N sets of data (essentially no limit on N), each consisting of one observation on each of n independent variables x_1 , x_2 , ..., x_n and a single dependent variable y which is coded as x_{m+1}.

The first part of the input deck consists of a set of comment cards headed by a card indicating the number of comment cards in the set. Any number of cards may be used for comments. The cards are read and punched directly so that comments will precede the program output when listed on the printer. This gives the user the opportunity to document the pertinent characteristics of each run. Comments may appear anywhere on the card. Blank cards for spacing purposes must also be counted. The count is punched in fixed point format in columns 1-5, and must be right justified.

Following the comment cards is a card containing the following (cf Section 3.3 for discussion of forms and procedures):

1. Columns 1-10, Format F10.5;

   Critical F-level for entering a term into the regression equation.

2. Columns 11-20, Format F10.5;

   Critical F-level for removing a term currently in the regression equation.

3. Columns 21-30, Format F10. 5;

   Value of standard error desired for the dependent variable. This accuracy value is used to determine when to terminate the computation. If no "feel" exists for this value, zero may be used in which case the computation must be terminated manually by use of Sense Switch 2 on the IBM 1620 console.

4. Columns 31-40, Format F10. 5;

   Constant term indicator. A "1" punch in column 32 indicates that a constant term should not be used in the regression equation. A "0" punch in column 32 indicates that a constant term should be used in the regression equation.

The next card contains the following information:

1. Columns 1-10, Format F10. 0;

   Number of terms to be considered per pass during the regression equation construction. (Maximum of 30.)

2. Columns 11-20, Format F10. 0;

   Maximum interaction order to be considered during term construction. (Must not be greater than the number of input variables less one.)

3. Columns 21-30, Format F10. 0;

   Interaction order weight indicator. A "1." indicates that all interaction order selection weights are initially equal. A "0." indicates that each interaction order has an associated selection weight to be read in as part of the input data.

4. Columns 31-40, Format F10. 0;

   Variable weight indicator. A "1. " indicates that all variable selection weights are initially equal. A "0. " indicates that each variable has an associated selection weight to be read in as part of the input data.

5. Columns 41-50, Format F10.0;

   Number of functions to be considered for each input variable during term construction. (Maximum of 20.)

6. Columns 51-60, Format F10.0;

   Function weight indicator. A "1." indicates that all function selection weights are initially equal. A "0." indicates that a set of function selection weights for each variable are to be used as part of the input data.

The next set of cards contains the data point information. The first card contains an ID number, the dependent variable,and up to six independent variables. The format used is 8F10.0. Additional independent variables are punched on succeeding cards in Columns 21-80. The last card of each data set need not be filled up.

If the interaction order weight indicator (see item 3 above) has been set to 0., the next set of cards contains the initial interaction order selection weights, up to eight per card, using the format 8F10.0. There must be as many weights as there are possible interaction orders as given by item 2 above. If the interaction order weight indicator has been set to 1., the initial selection weights will be set equal to each other automatically, obviating the interaction order weighting deck.

If the variable weight indicator (see item 4 above) has been set to 0., the next set of cards contains the initial variable selection weights, up to eight per card, using the format 8F10.0. There must be as many weights as there are input variables less 1, since the dependent variable is not weighted. If the variable weight indicator has been set to 1., the initial variable selection weights will be set equal to each other automatically, obviating the variable weighting deck.

If the function weight indicator (see item 6 above) has been set to 0., the next set of cards contains the initial function selection weights for each variable, up to eight per card, using the format 8F10.0. A set of initial function selection weights must be supplied for each of the independent variables. Sets must be arranged according to the order of the independent variables on the input data cards. The first card in each set contains the variable number in columns 1-10. (Format F10.0.) Each set must contain as many weights as there are possible functions specified (see item 5 above). If the function weight indicator has been set to 1., the initial selection weights will be set equal to each other automatically, obviating the function weighting deck.

The initial selection weights as discussed above need not be normalized by the user. This is done automatically. Setting any selection weight initially to zero completely rules out selection of that particular item for the duration of the analysis.

Should the user wish to terminate a SCORE run at any time, he may do so by turning Sense Switch 2 on. The current selector weights may be punched out by turning Sense Switch 1 on. Turning both switches on simultaneously will result in punching the selector weights and then terminating the run. These selector weights are punched in a format suitable for reading by the computer, should the user desire to continue the run at a latter time. Adjustment of the selector weight indicators as described above will allow for reading in these weights when beginning a subsequent run as a continuation of the same problem.

3.3 COMPUTER INPUT DATA TRANSMITTAL FORMS

Input Data Transmittal forms used to prepare data for a SCORE run on the computer are illustrated and explained in the following pages.

A. SCORE Data Transmittal Form I (DTF-I) (Figure E-7)

First Card

- NO. OF COMMENTS — The NO. OF COMMENTS card indicates the number of cards used for comments including blank cards to be read in and punched out at execution time.

Other Cards

- COMMENT CARDS — The COMMENT CARDS contain any information which the user desires to have punched out at the beginning of the program ouput listing. Such information as the name of the user, the date, and a description of the problem being submitted should be punched on these cards. Any legitimate alpha-numeric characters may be entered anywhere in columns 1-80. Output comment cards will have the identical format of the input comment cards.

LEAR SIEGLER, INC. INSTRUMENT DIVISION	SCORE DTF I	REGRESSION ANALYSIS	SHEET ______ OF ______

CARD 1

NO. COMMENTS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

COMMENT CARDS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

Figure E-7. SCORE Data Transmittal Form I (DTF-I)

B. SCORE Data Transmittal Form II (DTF-II) (Figure E-8)

First Card

- F-LEVEL IN — The F-LEVEL IN is the critical value of the F-level for entering a term into the regression equation.

- F-LEVEL OUT — The F-LEVEL OUT is the critical value of the F-level for removing a term from the regression equation. This value must be less than, or equal to the F-LEVEL IN value.

- STD. DEVIATION — The STD. DEVIATION is the accuracy factor desired for the regression equation. When the standard error of the dependent variable becomes less than this critical value, the run is terminated.

- CONSTANT — The CONSTANT entry is used to indicate whether or not a constant term should be used in the regression equation. A value of 0. indicates that a constant term should be used. A value of 1. indicates that a constant term should not be used.

- NO. VARIABLES — The NO. VARIABLES entry is the total number of input variables in the data set including the dependent variable.

- NO. DATA — The NO. DATA entry is the total number of data points in the data set being processed.

LSI	LEAR SIEGLER, INC. INSTRUMENT DIVISION	SCORE DTF II	REGRESSION ANALYSIS	SHEET ____ OF ____

CONTROL CARD 1

F - LEVEL IN	F - LEVEL OUT	STD. DEVIATION	CONSTANT	NO. VARIABLES	NO. DATA	
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

CONTROL CARD 2

NO. TERMS	INTERACTION ORDER	INTERACTION WEIGHT	VARIABLE WEIGHT	NO. FUNCTIONS	FUNCTION WEIGHT	
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

INPUT VARIABLES

IDENTIFICATION	DEPENDENT VAR.	INDEPENDENT VARIABLES (Fill only those used)						
1 2 3 4 5 6 7 8 9 10	11 12 13 14 15 16 17 18 19 20	21 22 23 24 25 26 27 28 29 30	31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70	71 72 73 74 75 76 77 78 79 80
	1	2	3	4	5	6	
	7	8	9	10	11	12	
	13	14	15	16	17	18	
	19	20	21	22	23	24	
	25	26	27	28			

Figure E-8. SCORE Data Transmittal Form II (DTF-II)

Second Card

- NO. TERMS — The NO. TERMS entry is the number of terms to be considered per pass during the regression equation construction.

- INTERACTION ORDER — The INTERACTION ORDER entry is the maximum interaction order to be considered during term construction. This value must not be greater than the number of input variables less one.

- INTERACTION WEIGHT — The INTERACTION WEIGHT entry is used to indicate whether or not interaction order selection weights will be supplied by the user. A value of 0. indicates that selection weights are to be read in as part of the input data. A value of 1. indicates that all interaction order selection weights are initially equal.

- VARIABLE WEIGHT — The VARIABLE WEIGHT entry is used to indicate whether or not variable selection weights will be supplied by the user. A value of 0. indicates that selection weights are to be read in as part of the input data. A value of 1. indicates that all variable selection weights are initially equal.

- NO. FUNCTIONS — The NO. FUNCTIONS entry is the maximum number of functions to be considered for each input variable during term construction.

- FUNCTION WEIGHT — The FUNCTION WEIGHT entry is used to indicate whether or not function selection weights will be supplied by the user. A value of 0. indicates that selection weights are to be read in as part of the input data. A value of 1. indicates that all function selection weights are initially equal.

Input Variable Card

- IDENTIFICATION — The IDENTIFICATION entry is an identification number assigned to each data point. This number is used for reference only. Any numbering sequence may be used.

- DEPENDENT VAR. — The DEPENDENT VAR. entry is the value of the dependent variable for each data point.

- INDEPENDENT VARIABLES — The INDEPENDENT VARIABLES area is reserved for the independent variables for each data point. As many cards as necessary may be used. Six variables per card are permitted.

C. SCORE Data Transmittal Form III (DTF-III) (Figure E-9)

Input Variable Card

This form is to be used for the data point variables in the data format described on DTF-II. Only as many cards as are necessary for each point should be used.

LEAR SIEGLER, INC. INSTRUMENT DIVISION	SCORE DTF III	REGRESSION ANALYSIS	SHEET ____ OF ____

INPUT VARIABLES (Continued)

IDENTIFICATION										DEPENDENT VAR.										INDEPENDENT VARIABLES (Fill only those used)																																																																		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

Figure E-9. SCORE Data Transmittal Form III (DTF-III)

D. SCORE Data Transmittal Form IV (DTF-IV) (Figure E-10)

- **INTERACTION ORDER SELECTION WEIGHTS** — If the Interaction Order Weight Indicator has been set to 0. (see DTF II, Second Card), the INTERACTION ORDER SELECTION WEIGHTS are entered in this area. There must be as many weights as there are possible interaction orders (see DTF II, Second Card).

- **VARIABLE SELECTION WEIGHTS** — If the Variable Weight Indicator has been set to 0. (see DTF II, Second Card), the VARIABLE SELECTION WEIGHTS are entered in this area. There must be as many weights as there are independent variables.

E. SCORE Data Transmittal Form V (DTF-V) (Figure E-11)

- **FUNCTION SELECTION WEIGHTS** — If the Function Weight Indicator has been set to 0. (see DTF II, Second Card), the FUNCTION SELECTION WEIGHTS for each variable are entered here. After the variable number is entered, it is followed by the selection weights. There must be as many selection weights for each variable as there are possible functions (see DTF II, Second Card).

3.4 LOGIC FLOW DIAGRAMS AND CORRESPONDING FORTRAN PROGRAM LISTINGS

This section contains a series of paired logic flow diagrams and FORTRAN program listings for the eight individual SCORE subprograms.

Figure E-12 shows a flow diagram of the SCORE mainline program logic. The program is composed almost entirely of interconnected subprograms (blocks labeled with the word SCORE and an accompanying number). A flow diagram of each of these eight subprograms is also included (see even numbered Figures E-14 through E-28).

3.4.1 SCORE Mainline Program

In addition to controlling the subprograms, the SCORE mainline program performs the following functions.

Upon the completion of each pass, information which consists of the actual value of the dependent variable, the predicted value of the dependent variable using the current regression equation, the deviation, or difference between these two values and the percent deviation, is punched out for each input data set.

The SCORE mainline program also checks to see whether the desired accuracy factor has been attained after each pass is completed. If so, the run is terminated.

The FORTRAN program listing for the SCORE mainline program is given in Figure E-13.

3.4.2 SCORE-1 Subprogram

Figure E-14 is a flow diagram for the SCORE-1 subprogram. This subprogram serves to read in the controlling variables for the equation generation and the input data sets along with their weights, if required, and punches these out for run documentation purposes. It can also read in the selection weights and normalizes them. Selection weights are stored on the magnetic disks.

The actual FORTRAN program listing for the SCORE -1 subprogram is given in Figure E-15.

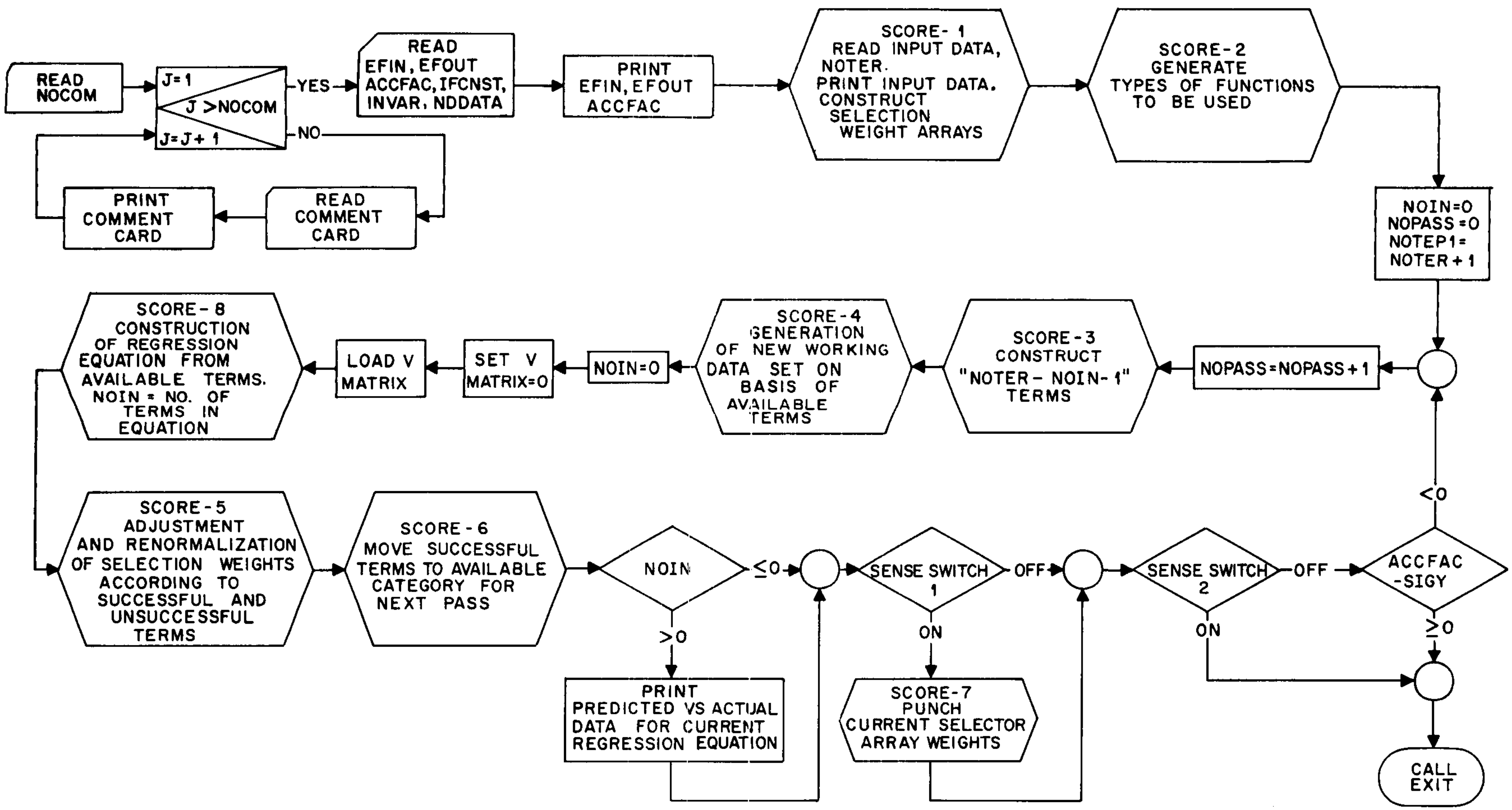


Figure E-12. SCORE Mainline Program -- Logic Flow Diagram

```
C
C                    *** DEFINITION OF VARIABLES ***
C
C      ACCFAC    STANDARD ERROR DESIRED FOR A SATISFACTORY REGRESSION RUN
C      CNST      CONSTANT TERM VALUE FOR REGRESSION EQUATION
C      COEN      REGRESSION EQUATION COEFFICIENT ARRAY
C      DATA      TEMPORARY STORAGE ARRAY OF DATA FOR EVALUATED TERMS
C      DEV       DEVIATION BETWEEN PREDICTED VALUE AND DATA VALUE OF DEPENDENT
C                VARIABLE
C      EFIN      CRITICAL F-LEVEL FOR ENTERING A TERM
C      EFOUT     CRITICAL F-LEVEL FOR REMOVING A TERM
C      FUN       ARRAY OF FUNCTIONS AVAILABLE FOR EACH VARIABLE
C      IBR       BRANCH INDICATOR
C      IDAVAI    DISC INDEX FOR LOCATING NEXT AVAILABLE AREA.
C      IDFN      DISC INDEX FOR LOCATING FUNCTION WEIGHTS
C      IDIOW     DISC INDEX FOR LOCATING INTERACTION ORDER WEIGHTS.
C      IDVARW    DISC INDEX FOR LOCATING VARIABLE WEIGHTS.
C      IFCNST    CONSTANT TERM INDICATOR. 1 IMPLIES NO CONSTANT TERM.
C                0 IMPLIES A CONSTANT TERM.
C      INDEX     ARRAY OF TERM NUMBERS IN THE REGRESSION EQUATION
C      INVAR     NUMBER OF INPUT VARIABLES INCLUDING THE DEPENDENT VARIABLE.
C      IVM1      NUMBER OF INPUT VARIABLES NOT INCLUDING THE DEPENDENT VARIABLE
C      MAXFUN    MAXIMUM NUMBER OF FUNCTIONS CONSIDERED
C      MAXIOR    MAXIMUM INTERACTION ORDER CONSIDERED.
C      NOCOM     NUMBER OF COMMENT CARDS TO BE READ AND PRINTED AT BEGINNING OF
C                RUN
C      NDATID    DISC INDEX FOR LOCATING DATA FOR EVAULATED TERMS
C      NODATA    NUMBER OF DATA SETS.
C      NOIN      NUMBER OF TERMS IN THE REGRESSION EQUATIONS
C      NOPASS    NUMBER OF PASSES ALREADY EXECUTED
C      NOTEP1    NUMBER OF TERMS CONSIDERED PER PASS PLUS ONE
C      NOTER     NUMBER OF TERMS CONSIDERED PER PASS INCLUDING THE DEPENDENT
C                VARIABLE
C      PERC      DEVIATION BETWEEN PREDICTED VALUE AND DATA VALUE OF DEPENDENT
C                VARIABLE IN PERCENT
C      RDPR      READ AND PRINT AREA FOR COMMENTS
C      RUN       DATA SET IDENTIFICATION NUMBER
C      SIGY      STANDARD ERROR FOR CURRENT REGRESSION EQUATION
C      V         REGRESSION MATRIX
C      YPRED     VALUE OF DEPENDENT VARIABLE USING REGRESSION EQUATION
```

Figure E-13. SCORE Mainline Program -- FORTRAN Program Listing (1 of 3)

```
C
C                                   *** PROGRAM ***
C
      DIMENSION DATA(30),V(31,31),IDFN(30),FUN(20),IDTER(30),RDPU(20),IN
     1DEX(30),COEN(30)
      COMMON V,COEN
  102 FORMAT(1H1,27X,19HSTEPWISE REGRESSION)
  103 FORMAT(20A4)
  104 FORMAT(I5)
  106 FORMAT(28H F LEVEL TO ENTER VARIABLE =,F10.6/)
  107 FORMAT(29H F LEVEL TO REMOVE VARIABLE =,F10.6/)
    6 FORMAT(34H DESIRED VALUE OF STANDARD ERROR =,F10.6///)
      XX=SQRTF(1.)
      READ 5,NOCOM
      DO 105 J=1,NOCOM
      READ 103,(RDPU(I),I=1,20)
  105 PRINT 103,(RDPU(I),I=1,20)
      R=RAND(R)
      VAR=ABSF(R)**1.5
      DEFINE DISK (10,1000)
  100 READ 5,EFIN,EFOUT,ACCFAC,IFCNST,INVAR,NODATA
    5 FORMAT(8F10.0)
C     IFCNST=1 DO NOT HAVE CONST TERM IN EQUATION
      PRINT 106,EFIN
      PRINT 107,EFOUT
      PRINT 6,ACCFAC
      CALL SCORE1(IDIOW,IDVARW,IDFN,IDAVAI,MAXFUN,NOTER,MAXIOR,IVM1,INVA
     1R,NODATA)
      CALL SCORE2(FUN)
      NOIN=0
      NOPASS=0
      NOTEP1=NOTER+1
  101 NOPASS=NOPASS+1
      CALL SCORE3(NOIN,NOTER,IDIOW,IDVARW,IDFN,IDAVAI,IVM1,MAXIOR,M
     1AXFUN,IDTER)
      CALL SCORE4(IDTER,IDAVAI,FUN,NOTER,INVAR,NODATA,NDATID,NOPASS)
      NOIN=0
      NOENT=0
  110 DO 120 I=1,NOTEP1
  130 DO 120 J=1,NOTEP1
  120 V(I,J)=0.
      ND=NDATID
  150 DO 170 N=1,NODATA
  160 FETCH(ND) RUN,(DATA(L),L=1,NOTER)
  180 DO 190 I=1,NOTER
  200 V(I,NOTEP1)=V(I,NOTEP1)+DATA(I)
  210 DO 220 J=I,NOTER
  220 V(I,J)=V(I,J)+DATA(I)*DATA(J)
  190 CONTINUE
```

Figure E-13. SCORE Mainline Program -- FORTRAN Program Listing (2 of 3)

```
  170 V(NOTEP1,NOTEP1)=V(NOTEP1,NOTEP1)+1.0
C     COMPLETED SUMS OF SQUARES AND CROSS PRODUCTS.  THESE ARE IN
C     STORAGE IN LOCATION,V(I,J)
  565 CALL SCORE8(NOTER,IFCNST,IBR,EFIN,EFOUT,INDEX,NOIN,CNST,ACCFAC,SIG
     1Y)
      IF(IBR)900,910,1581
 1581 CALL SCORE5(IDTER,IDIOW,IDVARW,IDFN,INDEX,NOIN,INVAR,MAXFUN,MAXIOR
     1,NOTER)
      CALL SCORE6(NOIN,INDEX,IDTER)
 1587 IF(NOIN)910,910,1583
 1583 PRINT 85
   85 FORMAT(//, 9X,36H        PREDICTED VS ACTUAL RESULTS //64H RUN NO.
     1     ACTUAL        PREDICTED      DEVIATION    PERCENT DEV.//)
      ND=NDATID
      FIND(ND)
 1590 DO 1660 N=1,NODATA
 1595 FETCH(ND) RUN,(DATA(L),L=1,NOTER)
 1610 YPRED=CNST
 1620 DO 1630 I=1,NOIN
 1640 K=INDEX(I)
 1630 YPRED=YPRED+COEN(I)*DATA(K)
 1650 DEV=YPRED-DATA(NOTER)
      IF(DATA(NOTER))1680,1670,1680
 1670 PRINT 80,RUN,DATA(NOTER),YPRED,DEV
      GO TO 1660
 1680 PERC=(DEV/ABS(DATA(NOTER)))*100.
      PRINT 80,RUN,DATA(NOTER),YPRED,DEV,PERC
 1660 CONTINUE
   80 FORMAT(F6.0,3X,E12.5,2X,E12.5,2X,E12.5,2X,E12.5)
C     SENSE SWITCH 1 ON - DUMP CURRENT SELECTOR WEIGHT ARRAYS
  910 IF(SENSE SWITCH 1)1588,1690
 1588 CALL SCORE7(IDIOW,MAXIOR,IDVARW,IVM1,MAXFUN,NOTER,IDFN)
 1690 IF(SENSE SWITCH 2)1700,1691
 1691 IF(ACCFAC-SIGY)101,1700,1700
 1700 CALL EXIT
  900 PRINT 905
  905 FORMAT (42H ERROR IN CONTROL CARD,PROBLEM TERMINATED )
      GO TO 910
      END
```

Figure E-13.

SCORE Mainline Program -- FORTRAN Program Listing (3 of 3)

ARGUMENTS	
INPUT	OUTPUT
INVAR	IDIOW
NODATA	IDVARW
	IDFN
	IDAVAI
	NOTER
	MAXFUN
	NOTER
	MAXIOR
	IVM1

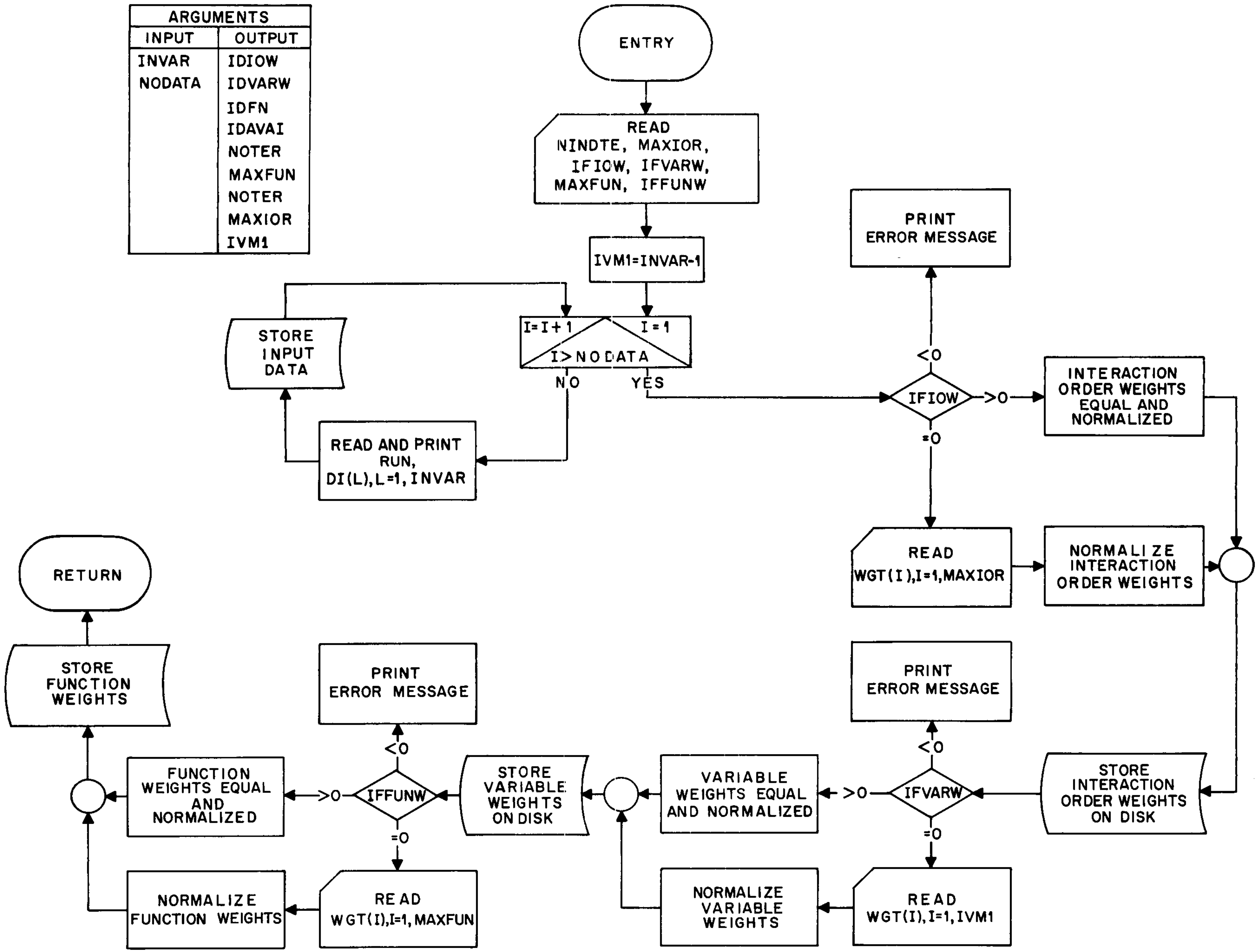


Figure E-14. SCORE-1 Subprogram -- Logic Flow Diagram

```
C
C                          *** DEFINITION OF VARIABLES ***
C
C     DI          TEMPORARY STORAGE ARRAY FOR INPUT DATA.
C     FIVM1       FLOATING POINT FORM OF THE NUMBER OF INPUT VARIABLES NOT
C                 INCLUDING THE DEPENDENT VARIABLE
C     FMAXIO      FLOATING POINT FORM OF THE MAXIMUM INTERACTION ORDER
C     FMAXFU      FLOATING POINT FORM OF THE MAXIMUM NUMBER OF FUNCTIONS BEING
C                 CONSIDERED.
C     IDAVAI      DISC INDEX FOR LOCATING NEXT AVAILABLE AREA.
C     IDFN        DISC INDEX FOR LOCATING FUNCTION WEIGHTS
C     IDIOW       DISC INDEX FOR LOCATING INTERACTION ORDER WEIGHTS.
C     IDVARW      DISC INDEX FOR LOCATING VARIABLE WEIGHTS.
C     IFFUNW      FUNCTION WEIGHT INDICATOR. 1 IMPLIES WEIGHTS READ IN.
C                 0 IMPLIES WEIGHTS INITIALLY EQUAL.
C     IFIOW       INTERACTION ORDER WEIGHT INDICATOR. 1 IMPLIES WEIGHTS READ IN.
C                 0 IMPLIES WEIGHTS INITIALLY EQUAL
C     IFVARW      VARIABLE WEIGHT INDICATOR. 1 IMPLIES WEIGHTS READ IN.
C                 0 IMPLIES WEIGHTS INITIALLY EQUAL
C     IVM1        NUMBER OF INPUT VARIABLES NOT INCLUDING THE DEPENDENT VARIABLE
C     K           VARIABLE NUMBER
C     MAXFUN      MAXIMUM NUMBER OF FUNCTIONS CONSIDERED
C     MAXIOR      MAXIMUM INTERACTION ORDER CONSIDERED.
C     NINDTE      NUMBER OF TERMS CONSIDERED PER PASS NOT INCLUDING THE DEPENDENT
C                 VARIABLE
C     NODATA      NUMBER OF DATA SETS.
C     NOTER       NUMBER OF TERMS CONSIDERED PER PASS INCLUDING THE DEPENDENT
C                 VARIABLE
C     RUN         DATA SET IDENTIFICATION NUMBER
C     SUMW        SUMMATION OF THE SELECTION WEIGHTS
C     WGT         TEMPORARY STORAGE ARRAY FOR SELECTION WEIGHTS
C
C                                  *** PROGRAM ***
C
C
C     INPUT DATA READ-IN AND PRINT-OUT
C
      SUBROUTINE SCORE1(IDIOW,IDVARW,IDFN,IDAVAI,MAXFUN,NOTER,MAXIOR,IVM
     11,INVAR,NODATA)
      DIMENSION DI(30),WGT(30),IDFN(30)
    6 FORMAT(10I5)
    7 FORMAT(16F5.0)
   10 FORMAT(8F10.0)
  100 FORMAT(20X,10HINPUT DATA)
  101 FORMAT(9H DATA SET,F6.0)
  102 FORMAT(4(1X,A2,I2,3H) =,E10.3))
  103 FORMAT(1X,8HWEIGHT =,F10.0//)
  104 FORMAT(3X,5HY    =,E10.3)
      ITRICK=6724
      MD=1
      FIND (MD)
      READ 10,NINDTE,MAXIOR,IFIOW,IFVARW,MAXFUN,IFFUNW
      NOTER=NINDTE+1
      IVM1=INVAR-1
```

Figure E-15. SCORE-1 Subprogram -- FORTRAN Program Listing (1 of 3)

```
      DO 1 I=1,NODATA
    3 IF(INVAR-7)2,2,30
   30 READ 10, RUN,DI(INVAR),(DI(L),L=1,6)
      READ 105,(DI(L),L=7,IVM1)
  105 FORMAT(20X,6F10.0)
      GO TO 4
    2 READ 10,RUN,DI(INVAR),(DI(L),L=1,IVM1)
    4 PRINT 101,RUN
      PRINT 102,(ITRICK,L,DI(L),L=1,IVM1)
      PRINT 104,DI(INVAR)
      RECORD(MD) RUN,(DI(L),L=1,INVAR)
    1 CONTINUE
      IDIOW=MD
      IF(IFIOW)5,28,11
    5 PRINT 40
      CALL EXIT
   40 FORMAT(42H ERROR IN CONTROL CARD, PROBLEM TERMINATED)
   28 READ 10,(WGT(I),I=1,MAXIOR)
C     READ INTERACTION ORDER WEIGHTS.
      SUMW=0.
      DO 8 I=1,MAXIOR
    8 SUMW=SUMW+WGT(I)
      DO 9 I=1,MAXIOR
    9 WGT(I)=WGT(I)/SUMW
C     INTERACTION ORDER WEIGHTS NORMALIZED.
      GO TO 12
   11 FMAXIO=MAXIOR
      DO 13 I=1,MAXIOR
   13 WGT(I)=1./FMAXIO
   12 RECORD (MD) (WGT(I),I=1,MAXIOR)
      IDVARW=MD
      IF(IFVARW)5,14,15
   14 READ 10,(WGT(I),I=1,IVM1)
C     READ INPUT VARIABLE WEIGHTS.
      SUMW=0.
      DO 16 I=1,IVM1
   16 SUMW=SUMW+WGT(I)
      DO 17 I=1,IVM1
   17 WGT(I)=WGT(I)/SUMW
C     VARIABLE SELECTION WEIGHTS NORMALIZED.
      GO TO 18
   15 FIVM1=IVM1
      DO 19 I=1,IVM1
   19 WGT(I)=1./FIVM1
   18 RECORD(MD) (WGT(I),I=1,IVM1)
      IDFN(1)=MD
      IF(IFFUNW)5,20,21
   20 DO 29 J=1,IVM1
      READ 10,K
      READ 10,(WGT(I),I=1,MAXFUN)
C     READ FUNCTION WEIGHTS.
      SUMW=0.
```

Figure E-15. SCORE-1 Subprogram -- FORTRAN Program Listing (2 of 3)

```
      DO 22 I=1,MAXFUN
   22 SUMW=SUMW+WGT(I)
      DO 23 I=1,MAXFUN
   23 WGT(I)=WGT(I)/SUMW
C     FUNCTION WEIGHTS NORMALIZED
      RECORD(MD)(WGT(I),I=1,MAXFUN)
   29 IDFN(J+1)=MD
      IDAVAI=MD
      RETURN
   21 FMAXFU=MAXFUN
      DO 25 I=1,MAXFUN
   25 WGT(I)=1./FMAXFU
   24 DO 26 J=1,IVM1
      RECORD(MD) (WGT(I),I=1,MAXFUN)
   26 IDFN(J+1)=MD
      IDAVAI=MD
      RETURN
      END
```

Figure E-15. SCORE-1 Subprogram -- FORTRAN Program Listing (3 of 3)

3.4.3 SCORE-2 Subprogram

Figure E-16 is a flow diagram for the SCORE-2 subprogram. This subprogram sets up the functions of the input variables which are to be considered during term construction. The FORTRAN program listing for SCORE-2 is given in Figure E-17.

3.4.4 SCORE-3 Subprogram

Figure E-18 is a flow diagram for the SCORE-3 subprogram. This subprogram constructs the terms to be considered in the stepwise regression process. This construction is based on a weighted random selection as discussed in the preceding text. Each new term is checked to make sure it does not duplicate another term previously constructed. The FORTRAN program listing for SCORE-3 is given in Figure E-19.

3.4.5 SCORE-4 Subprogram

Figure E-20 is a flow diagram for the SCORE-4 subprogram. This subprogram, evaluating the terms constructed in the SCORE-3 subprogram at each of the data sets, effectively forms a new data set. This data is then stored on the magnetic disks. The FORTRAN program listing for SCORE-4 is given in Figure E-21.

3.4.6 SCORE-5 Subprogram

Figure E-22 is a flow diagram for the SCORE-5 subprogram. This subprogram is used to adjust the selector weights on the basis of those terms which successfully entered the regression equation and those terms which were available but were not successful in entering the regression equation. After all adjustments are made according to success and failure, the selection weights are normalized. The FORTRAN program listing for SCORE-5 is given in Figure E-23.

3.4.7 SCORE-6 Subprogram

Figure E-24 is a flow diagram for the SCORE-6 subprogram. This subprogram is used to rearrange the successful terms of the previous pass so that they are retained as available terms for the next pass. The FORTRAN program listing for SCORE-6 is given in Figure E-25.

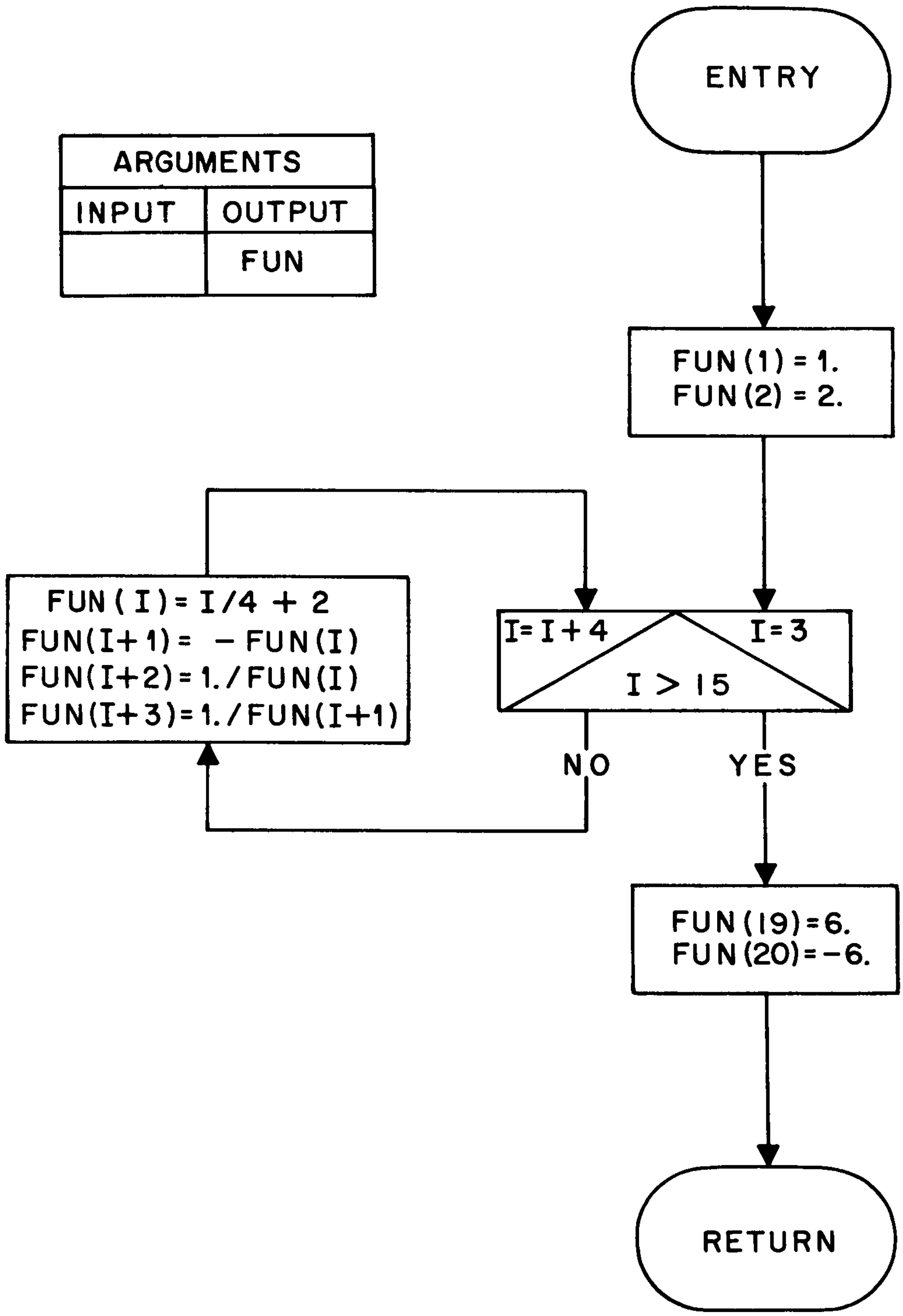


Figure E-16. SCORE-2 Subprogram -- Logic Flow Diagram

SCORE-2 SUBPROGRAM LISTING

```
C
C                         *** DEFINITION OF VARIABLES ***
C
C     FUN           ARRAY OF FUNCTIONS AVAILABLE FOR EACH VARIABLE
C
C                                  *** PROGRAM ***
C
      SUBROUTINE SCORE2(FUN)
      DIMENSION FUN(20)
      FUN(1)=1.
      FUN(2)=-1.
      DO 1 I=3,15,4
      FUN(I)=I/4+2
      FUN(I+1)=-FUN(I)
      FUN(I+2)=1./FUN(I)
    1 FUN(I+3)=1./FUN(I+1)
      FUN(19)=6.
      FUN(20)=-6.
      RETURN
      END
```

Figure E-17. SCORE-2 Subprogram -- FORTRAN Program Listing

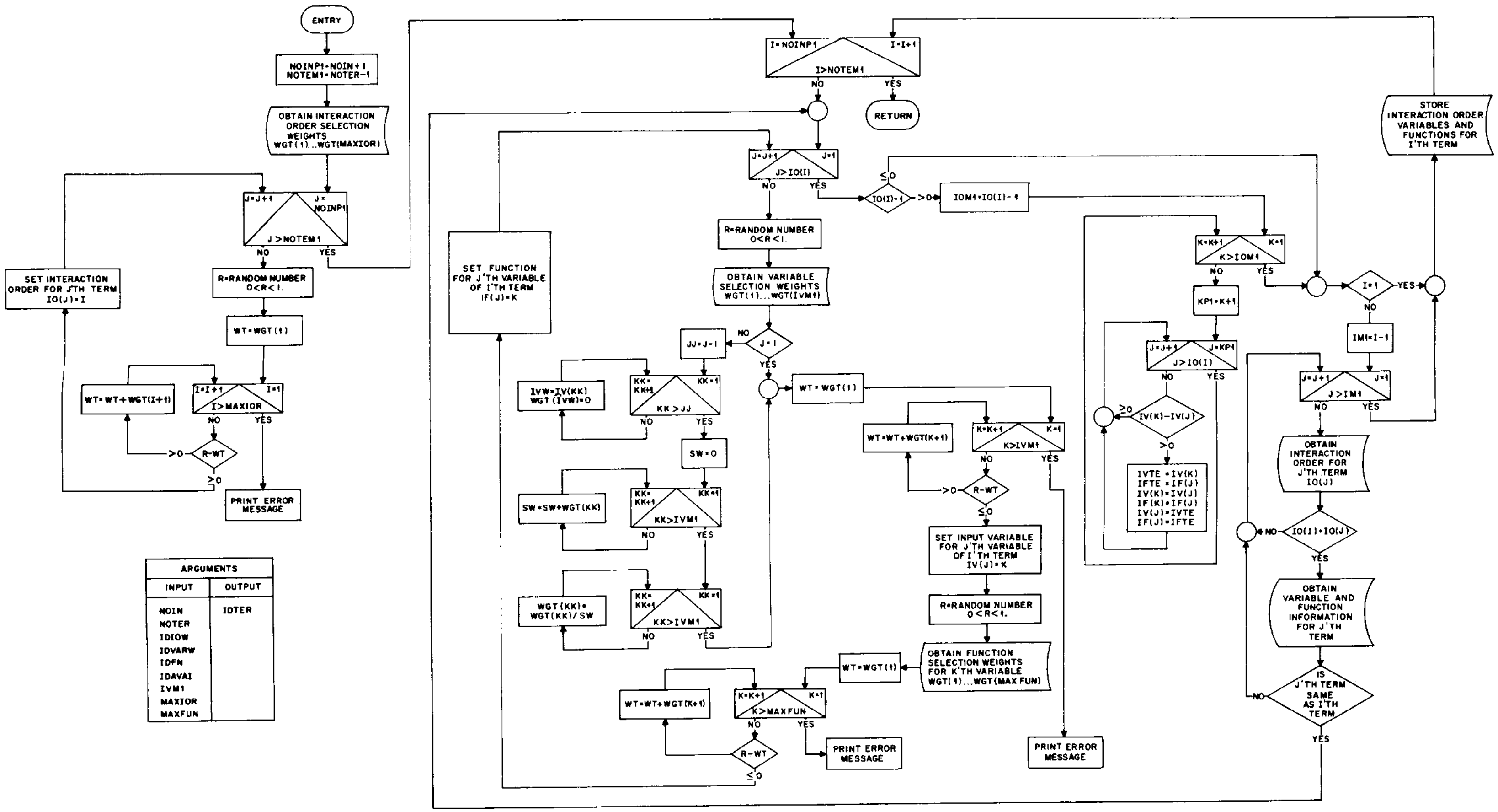


Figure E-18. SCORE-3 Subprogram -- Logic Flow Diagram

```
C
C                        *** DEFINITION OF VARIABLES ***
C
C     IDAVAI      DISC INDEX FOR LOCATING NEXT AVAILABLE AREA.
C     IDFN        DISC INDEX FOR LOCATING FUNCTION WEIGHTS
C     IDIOW       DISC INDEX FOR LOCATING INTERACTION ORDER WEIGHTS.
C     IDTER       DISC INDEX FOR LOCATING TERM DESCRIPTIONS
C     IDVARW      DISC INDEX FOR LOCATING VARIABLE WEIGHTS.
C     IF          FUNCTION NUMBER ARRAY
C     IFT         TEMPORARY STORAGE OF FUNCTION NUMBER
C     IFTE        TEMPORARY STORAGE OF FUNCTION NUMBER
C     IO          INTERACTION ORDER ARRAY OF CURRENT AVAILABLE TERMS
C     IOM1        CURRENT INTERACTION ORDER MINUS ONE
C     IOW         WORKING VALUE OF THE INTERACTION ORDER
C     IV          VARIABLE NUMBER ARRAY FOR CURRENT AVAILABLE TERMS
C     IVM1        NUMBER OF INPUT VARIABLES NOT INCLUDING THE DEPENDENT VARIABLE
C     IVT         TEMPORARY STORAGE OF VARIABLE NUMBER
C     IVTE        TEMPORARY STORAGE OF VARIABLE NUMBER
C     IVW         WORKING VALUE OF VARIABLE NUMBER
C     MAXFUN      MAXIMUM NUMBER OF FUNCTIONS CONSIDERED
C     MAXIOR      MAXIMUM INTERACTION ORDER CONSIDERED.
C     NOIN        NUMBER OF TERMS IN THE REGRESSION EQUATION
C     NOINP1      NUMBER OF TERMS IN THE REGRESSION EQUATION PLUS ONE
C     NOTEM1      NUMBER OF TERMS CONSIDERED PER PASS MINUS ONE
C     NOTER       NUMBER OF TERMS CONSIDERED PER PASS INCLUDING THE DEPENDENT
C                 VARIABLE
C     R           RANDOM NUMBER BETWEEN 0 AND 1.
C     SW          SUMMATION OF SELECTION WEIGHTS.
C     WGT         TEMPORARY STORAGE ARRAY FOR SELECTION WEIGHTS
C     WT          PARTIAL SUMMATION OF SELECTION WEIGHTS
C
C                                 *** PROGRAM ***
C
      SUBROUTINE SCORE3(NOIN,NOTER,IDIOW,IDVARW,IDFN,IDAVAI,IVM1,MA
     1XIOR,MAXFUN,IDTER)
      DIMENSION IDFN(30),WGT(30),IO(30),IDTER(30),IV(30),IF(30),IVT(30)
     1,IFT(30)
    6 FORMAT(14I5)
  100 FORMAT(F10.6)
      IDTER(1)=IDFN(IVM1+1)
      NOINP1=NOIN+1
      NOTEM1=NOTER-1
      MD=IDIOW
      FETCH(MD)(WGT(I),I=1,MAXIOR)
      DO 3 J=NOINP1,NOTEM1
      R=RAND(R)
      WT=WGT(1)
      DO 1 I=1,MAXIOR
      IF(R-WT)3,3,2
    2 WT=WT+WGT(I+1)
    1 CONTINUE
    3 IO(J)=I
```

Figure E-19. SCORE-3 Subprogram -- FORTRAN Program Listing (1 of 3)

```
      DO 4 I=NOINP1,NOTEM1
   22 IOW=IO(I)
      DO 5 J=1,IOW
      MD=IDVARW
      R=RAND(R)
      FETCH(MD)(WGT(K),K=1,IVM1)
      IF(J-1)12,12,13
   13 JJ=J-1
      DO 14 KK=1,JJ
      IVW=IV(KK)
   14 WGT(IVW)=0.
      SW=0.
      DO 15 KK=1,IVM1
   15 SW=SW+WGT(KK)
      DO 16 KK=1,IVM1
   16 WGT(KK)=WGT(KK)/SW
   12 WT=WGT(1)
      DO 26 K=1,IVM1
      IF(R-WT)7,7,8
    8 WT=WT+WGT(K+1)
   26 CONTINUE
    7 IV(J)=K
      MD=IDFN(K)
      R=RAND(R)
      FETCH(MD)(WGT(K),K=1,MAXFUN)
      WT=WGT(1)
      DO 9 K=1,MAXFUN
      IF(R-WT)10,10,11
   11 WT=WT+WGT(K+1)
    9 CONTINUE
   10 IF(J)=K
    5 CONTINUE
      IF(IO(I)-1)30,30,31
   31 IOM1=IO(I)-1
      DO 32 K=1,IOM1
      KP1=K+1
      DO 32 J=KP1,IOW
      IF(IV(K)-IV(J))32,32,33
   33 IVTE=IV(K)
      IFTE=IF(K)
      IV(K)=IV(J)
      IF(K)=IF(J)
      IV(J)=IVTE
      IF(J)=IFTE
   32 CONTINUE
C
C     CHECK FOR DUPLICATION OF PREVIOUS TERM
C
   30 IF(I-1)23,23,24
   24 IM1=I-1
      DO 18 J=1,IM1
      MD=IDTER(J)
```

Figure E-19. SCORE-3 Subprogram -- FORTRAN Program Listing (2 of 3)

```
      FETCH(MD) IO(J)
      IF(IO(I)-IO(J))18,19,18
   19 MD=IDTER(J)
      FETCH(MD)IO(J),(IVT(K),IFT(K),K=1,IOW)
      DO 20 K=1,IOW
      IF(IV(K)-IVT(K))18,21,18
   21 IF(IF(K)-IFT(K))18,20,18
   20 CONTINUE
      GO TO 22
   18 CONTINUE
   23 MD=IDTER(I)
      FIND(MD)
      RECORD(MD)IO(I),(IV(K),IF(K),K=1,IOW)
      IDTER(I+1)=MD
    4 CONTINUE
      IDAVAI=MD
      RETURN
      END
```

Figure E-19. SCORE-3 Subprogram -- FORTRAN Program Listing (3 of 3)

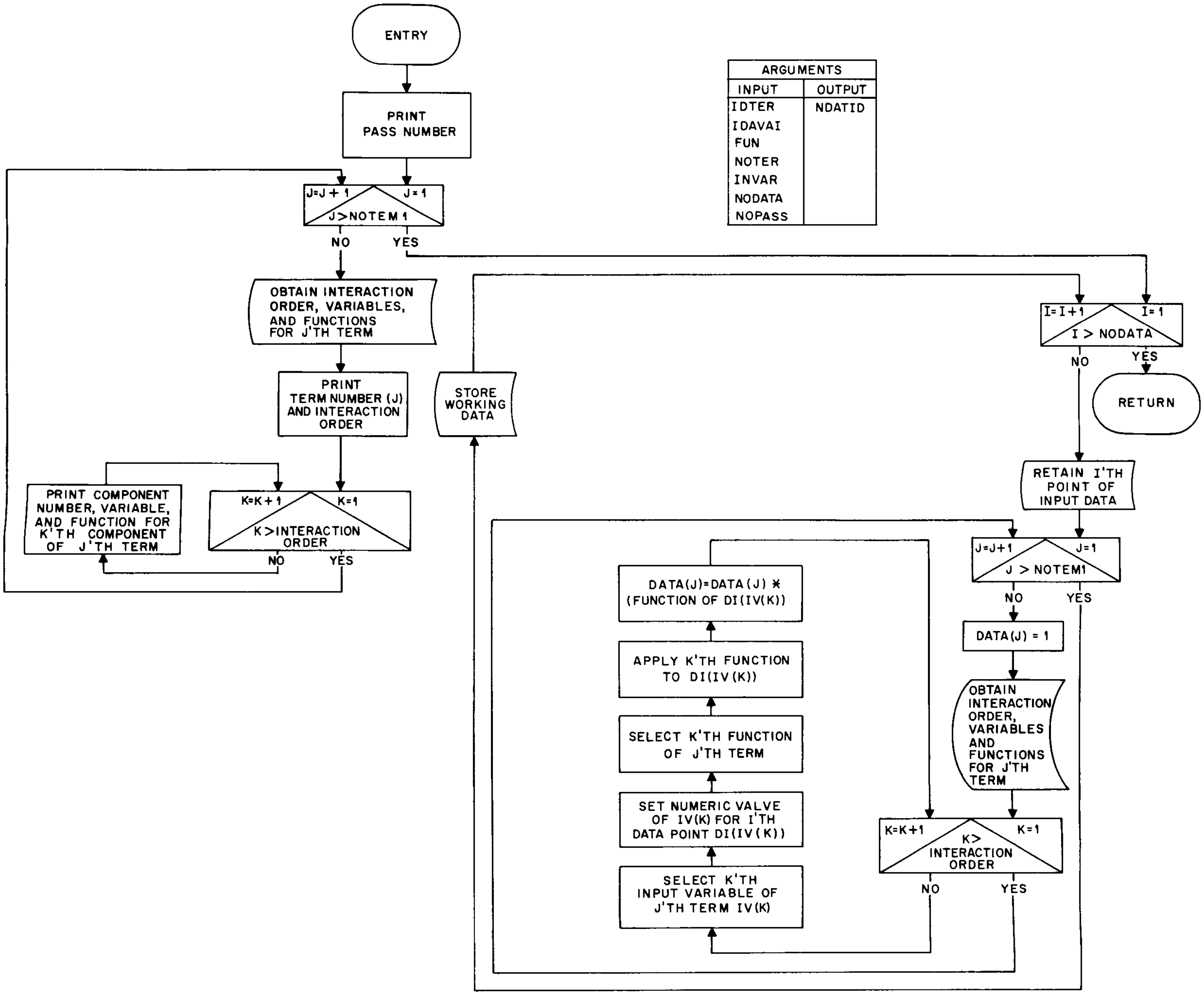


Figure E-20. SCORE-4 Subprogram -- Logic Flow Diagram

```
C
C                              *** DEFINITION OF VARIABLES ***
C
C     DATA        TEMPORARY STORAGE ARRAY OF DATA FOR EVALUATED TERMS
C     DI          TEMPORARY STORAGE ARRAY FOR INPUT DATA.
C     FUN         ARRAY OF FUNCTIONS AVAILABLE FOR EACH VARIABLE
C     FUNW        WORKING VALUE OF FUNCTION BEING APPLIED TO VARIABLE
C     IDAVAI      DISC INDEX FOR LOCATING NEXT AVAILABLE AREA.
C     IDTER       DISC INDEX FOR LOCATING TERM DESCRIPTIONS
C     IF          FUNCTION NUMBER ARRAY
C     IFUNW       FIXED POINT FORM OF THE WORKING VALUE OF THE FUNCTION
C     IFW         WORKING VALUE OF FUNCTION NUMBER
C     INVAR       NUMBER OF INPUT VARIABLES INCLUDING THE DEPENDENT VARIABLE.
C     IO          INTERACTION ORDER ARRAY OF CURRENT AVAILABLE TERMS
C     IOW         WORKING VALUE OF THE INTERACTION ORDER
C     IV          VARIABLE NUMBER ARRAY FOR CURRENT AVAILABLE TERMS
C     IVW         WORKING VALUE OF VARIABLE NUMBER
C     NDATID      DISC INDEX FOR LOCATING DATA FOR EVAULATED TERMS
C     NODATA      NUMBER OF DATA SETS.
C     NOPASS      NUMBER OF PASSES ALREADY EXECUTED
C     NOTEM1      NUMBER OF TERMS CONSIDERED PER PASS MINUS ONE
C     NOTER       NUMBER OF TERMS CONSIDERED PER PASS INCLUDING THE DEPENDENT
C                 VARIABLE
C     RUN         DATA SET IDENTIFICATION NUMBER
C
C                                      *** PROGRAM ***
C
C
C     EVALUATION OF TERMS
C
      SUBROUTINE SCORE4(IDTER,IDAVAI,FUN,NOTER,INVAR,NODATA,NDATID,
     1NOPASS)
      DIMENSION IDTER(30),DI(30),FUN(20),DATA(30),IV(30),IF(30)
    6 FORMAT(I5)
  100 FORMAT(/1X,5HTERM(,I2,25H) = INTERACTION OF ORDER ,I2,6H WITH,)
  101 FORMAT(11X,10HCOMPONENT(,I2,6H) = X(,I2,4H)**(,F6.3,1H))
  102 FORMAT(1H1,/29X,12HPASS NUMBER ,I3,//1X,17HTERM DESCRIPTIONS)
      PRINT 102,NOPASS
      MD=1
      ND=IDAVAI
      NDATID=ND
      NOTEM1=NOTER-1
      DO 10 J=1,NOTEM1
      LD=IDTER(J)
      FETCH(LD)IO
      IOW=IO
      LD=IDTER(J)
      FETCH(LD) IO,(IV(L),IF(L),L=1,IOW)
      PRINT 100,J,IO
      DO 11 K=1,IO
      IVW=IV(K)
      IFW=IF(K)
      FUNW=FUN(IFW)
```

Figure E-21. SCORE-4 Subprogram -- FORTRAN Program Listing (1 of 2)

```
11 PRINT 101,K,IVW,FUNW
10 CONTINUE
   DO 1 I=1,NODATA
   FIND (MD)
 3 FETCH(MD)RUN,(DI(L),L=1,INVAR)
 4 DO 5 J=1,NOTEM1
   DATA(J)=1.
   LD=IDTER(J)
   FIND(LD)
   FETCH(LD)IO
   IOW=IO
   LD=IDTER(J)
   FIND(LD)
   FETCH(LD) IO,(IV(L),IF(L),L=1,IOW)
   DO 26 K=1,IO
   IVW=IV(K)
   IFW=IF(K)
   FUNW=FUN(IFW)
   IF(ABSF(FUNW)-1.)12,13,13
13 IFUNW=FUNW
   DATA(J)=DATA(J)*DI(IVW)**IFUNW
   GO TO 26
12 DATA(J)=DATA(J)*DI(IVW)**FUNW
26 CONTINUE
 5 CONTINUE
   DATA(NOTER)=DI(INVAR)
   FIND (ND)
 8 RECORD(ND)RUN,(DATA(L),L=1,NOTER)
 1 CONTINUE
   IDAVAI=ND
   RETURN
   END
```

Figure E-21. SCORE-4 Subprogram -- FORTRAN Program Listing (2 of 2)

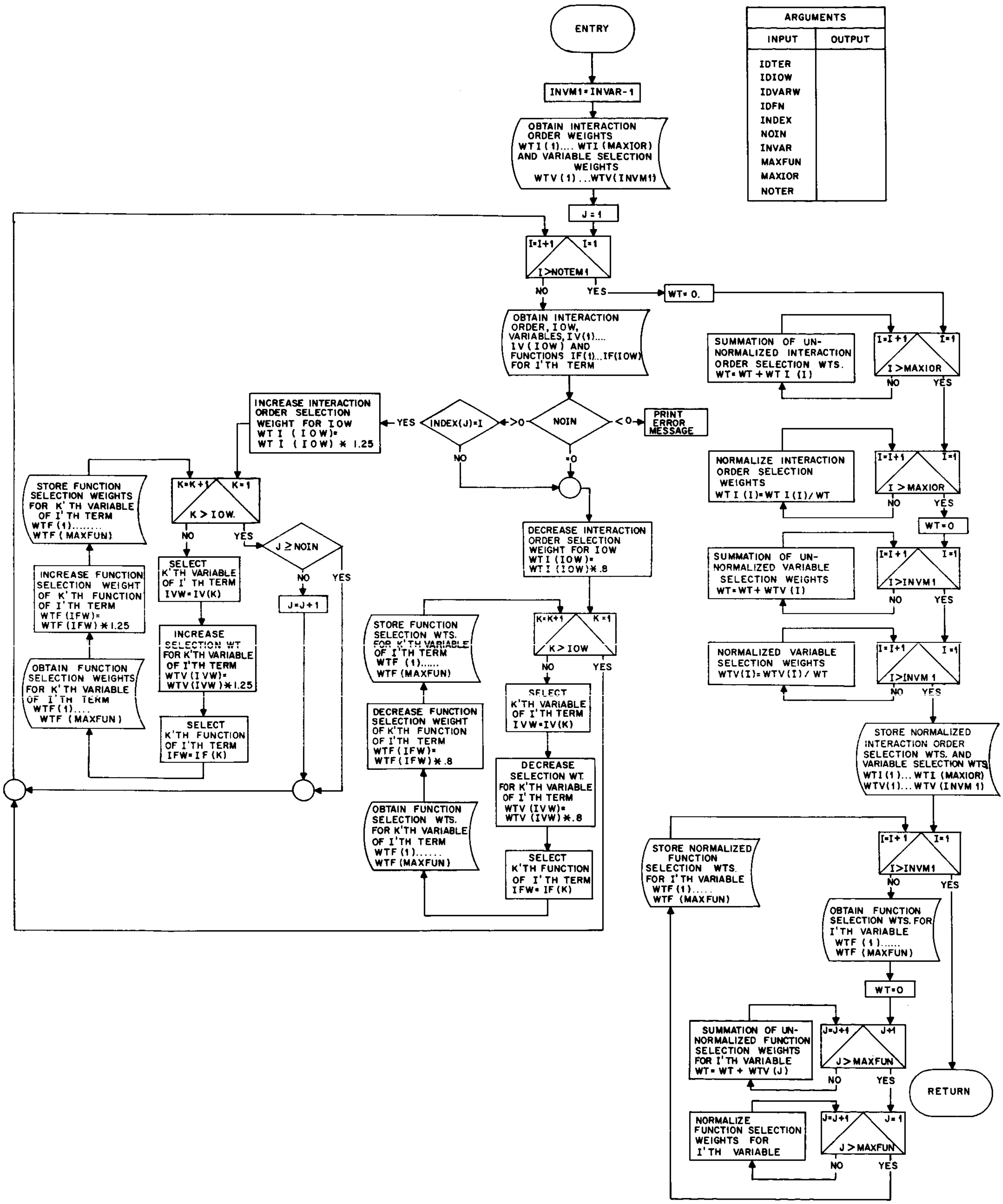


Figure E-22. SCORE-5 Subprogram -- Logic Flow Diagram

```
C
C                    *** DEFINITION OF VARIABLES ***
C
C     IDFN        DISC INDEX FOR LOCATING FUNCTION WEIGHTS
C     IDIOW       DISC INDEX FOR LOCATING INTERACTION ORDER WEIGHTS.
C     IDTER       DISC INDEX FOR LOCATING TERM DESCRIPTIONS
C     IDVARW      DISC INDEX FOR LOCATING VARIABLE WEIGHTS.
C     IFW         WORKING VALUE OF FUNCTION NUMBER
C     INDEX       ARRAY OF TERM NUMBERS IN THE REGRESSION EQUATION
C     INVAR       NUMBER OF INPUT VARIABLES INCLUDING THE DEPENDENT VARIABLE.
C     INVM1       NUMBER OF INPUT VARIABLES NOT INCLUDING THE DEPENDENT VARIABLE
C     IVW         WORKING VALUE OF VARIABLE NUMBER
C     MAXFUN      MAXIMUM NUMBER OF FUNCTIONS CONSIDERED
C     MAXIOR      MAXIMUM INTERACTION ORDER CONSIDERED.
C     NOIN        NUMBER OF TERMS IN THE REGRESSION EQUATION
C     NOTEM1      NUMBER OF TERMS CONSIDERED PER PASS MINUS ONE
C     NOTER       NUMBER OF TERMS CONSIDERED PER PASS INCLUDING THE DEPENDENT
C                 VARIABLE
C     WT          SUMMATION OF SELECTION WEIGHTS
C     WTF         FUNCTION SELECTION WEIGHT ARRAY
C     WTI         INTERACTION ORDER WEIGHT ARRAY.
C     WTV         VARIABLE WEIGHT ARRAY
C
C                              *** PROGRAM ***
C
      SUBROUTINE SCORE5(IDTER,IDIOW,IDVARW,IDFN,INDEX,NOIN,INVAR,MAXFUN,
     1MAXIOR,NOTER)
      DIMENSION IDTER(30),IDFN(30),INDEX(30),WTI(30),WTV(30),IV(30),IF(3
     10),WTF(30)
      NOTEM1=NOTER-1
      INVM1=INVAR-1
      MD=IDIOW
      FIND(MD)
      FETCH(MD)(WTI(I),I=1,MAXIOR)
      FETCH(MD)(WTV(I),I=1,INVM1)
      J=1
      DO 1 I=1,NOTEM1
      MD=IDTER (I)
      FETCH(MD)IOW
      MD=IDTER(I)
      FETCH(MD) IO,(IV(K),IF(K),K=1,IOW)
      IF(NOIN)3,3,16
   16 IF(INDEX(J)-I)3,4,3
    4 WTI(IOW)=WTI(IOW)*1.25
      DO 5 K=1,IOW
      IVW=IV(K)
      WTV(IVW)=WTV(IVW)*1.25
      IFW=IF(K)
      MD=IDFN(IVW)
      FETCH(MD)(WTF(L),L=1,MAXFUN)
      WTF(IFW)=WTF(IFW)*1.25
      MD=IDFN(IVW)
```

Figure E-23. SCORE-5 Subprogram -- FORTRAN Program Listing (1 of 2)

```
    5 RECORD(MD)(WTF(L),L=1,MAXFUN)
      IF(J-NOIN)7,1,1
    7 J=J+1
      GO TO 1
    3 WTI(IOW)=WTI(IOW)*.8
      DO 2 K=1,IOW
      IVW=IV(K)
      WTV(IVW)=WTV(IVW)*.8
      IFW=IF(K)
      MD=IDFN(IVW)
      FETCH(MD)(WTF(L),L=1,MAXFUN)
      WTF(IFW)=WTF(IFW)*.8
      MD=IDFN(IVW)
    2 RECORD(MD)(WTF(L),L=1,MAXFUN)
    1 CONTINUE
      WT=0.
      DO 8 I=1,MAXIOR
    8 WT=WT+WTI(I)
      DO 9 I=1,MAXIOR
    9 WTI(I)=WTI(I)/WT
C     INTERACTION WEIGHTS NORMALIZED
      WT=0.
      DO 10 I=1,INVM1
   10 WT=WT+WTV(I)
      DO 11 I=1,INVM1
   11 WTV(I)=WTV(I)/WT
C     VARIABLE WEIGHTS NORMALIZED
      MD=IDIOW
      RECORD(MD)(WTI(I),I=1,MAXIOR)
      RECORD(MD)(WTV(I),I=1,INVM1)
      DO 12 I=1,INVM1
      MD=IDFN(I)
      FETCH(MD)(WTF(K),K=1,MAXFUN)
      WT=0.
      DO 13 J=1,MAXFUN
   13 WT=WT+WTF(J)
      DO 14 J=1,MAXFUN
   14 WTF(J)=WTF(J)/WT
      MD=IDFN(I)
      RECORD(MD)(WTF(K),K=1,MAXFUN)
   12 CONTINUE
C     FUNCTION WEIGHTS NORMALIZED
      RETURN
      END
```

Figure E-23. SCORE-5 Subprogram -- FORTRAN Program Listing (2 of 2)

ARGUMENTS	
INPUT	OUTPUT
NOIN INDEX IDTER	

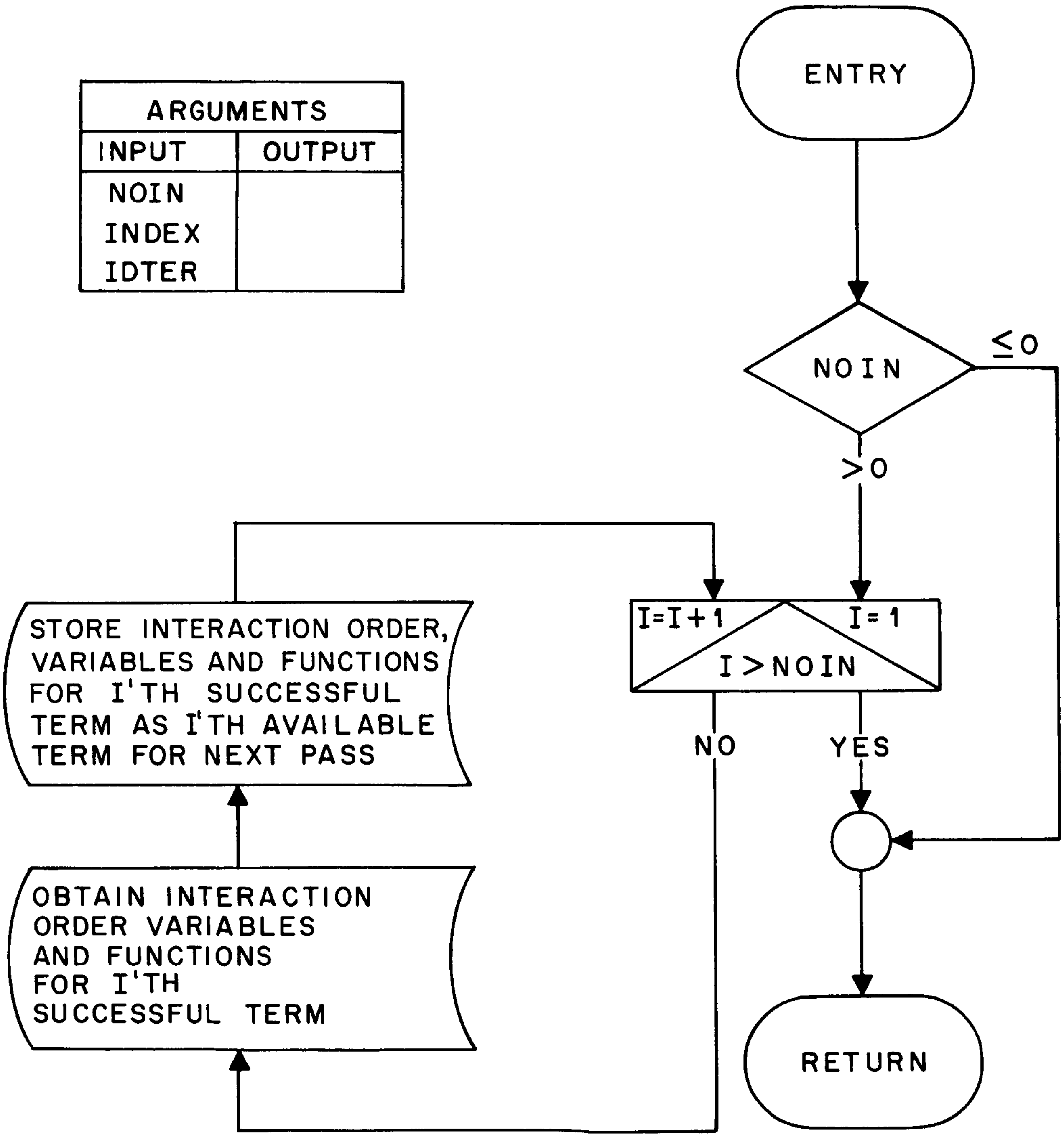


Figure E-24. SCORE-6 Subprogram -- Logic Flow Diagram

FIGURE E-25
SCORE-6 SUBPROGRAM LISTING

```
C
C                          *** DEFINITION OF VARIABLES ***
C
C     IDTER          DISC INDEX FOR LOCATING TERM DESCRIPTIONS
C     IF             FUNCTION NUMBER ARRAY
C     IND            TEMPORARY VALUE OF TERM NUMBER
C     INDEX          ARRAY OF TERM NUMBERS IN THE REGRESSION EQUATION
C     IOW            WORKING VALUE OF THE INTERACTION ORDER
C     IV             VARIABLE NUMBER ARRAY FOR CURRENT AVAILABLE TERMS
C     NOIN           NUMBER OF TERMS IN THE REGRESSION EQUATION
C
C                                  *** PROGRAM ***
C
      SUBROUTINE SCORE6(NOIN,INDEX,IDTER)
      DIMENSION INDEX(30),IDTER(30),IV(30),IF(30)
      ND=IDTER(1)
      IF(NOIN)2,2,3
    3 DO 1 I=1,NOIN
      IND=INDEX(I)
      MD=IDTER(IND)
      FETCH(MD)IOW
      MD=IDTER(IND)
      FETCH(MD)IO,(IV(J),IF(J),J=1,IOW)
      RECORD(ND)IO,(IV(J),IF(J),J=1,IO)
      IDTER(I+1)=ND
    1 CONTINUE
    2 RETURN
      END
```

Figure E-25. SCORE-6 Subprogram -- FORTRAN Program Listing

3.4.8 SCORE-7 Subprogram

Figure E-26 is a flow diagram for the SCORE-7 subprogram. This subprogram punches out the selector weights in a format suitable for inputting them again later if desired. This subprogram is activated from the SCORE mainline program by turning the IBM 1620 Sense Switch 1 on. The FORTRAN program listing for SCORE-7 is given in Figure E-27.

3.4.9 SCORE-8 Subprogram

Figure E-28 is a flow diagram for the SCORE-8 subprogram. This subprogram comprises the "meat" of the entire network of SCORE subprograms. Actual construction of the regression equation takes place here along with the computation of the standard error of y, the coefficient of multiple correlation and the coefficient of determination as described previously in this report. In addition, the least squares coefficients of each term in the regression equation and the standard error of these coefficients are computed. The step-by-step process of equation construction is punched out in this subprogram. The FORTRAN program listing for SCORE-8 is given in Figure E-29.

For those interested in investigating the program operation further, a list of variable names used in the FORTRAN programs along with a brief definition of each is given at the beginning of each of the program listings.

ARGUMENTS	
INPUT	OUTPUT
IDIOW	
MAXIOR	
IDVARW	
IVM1	
MAXFUN	
NOTER	
IDFN	

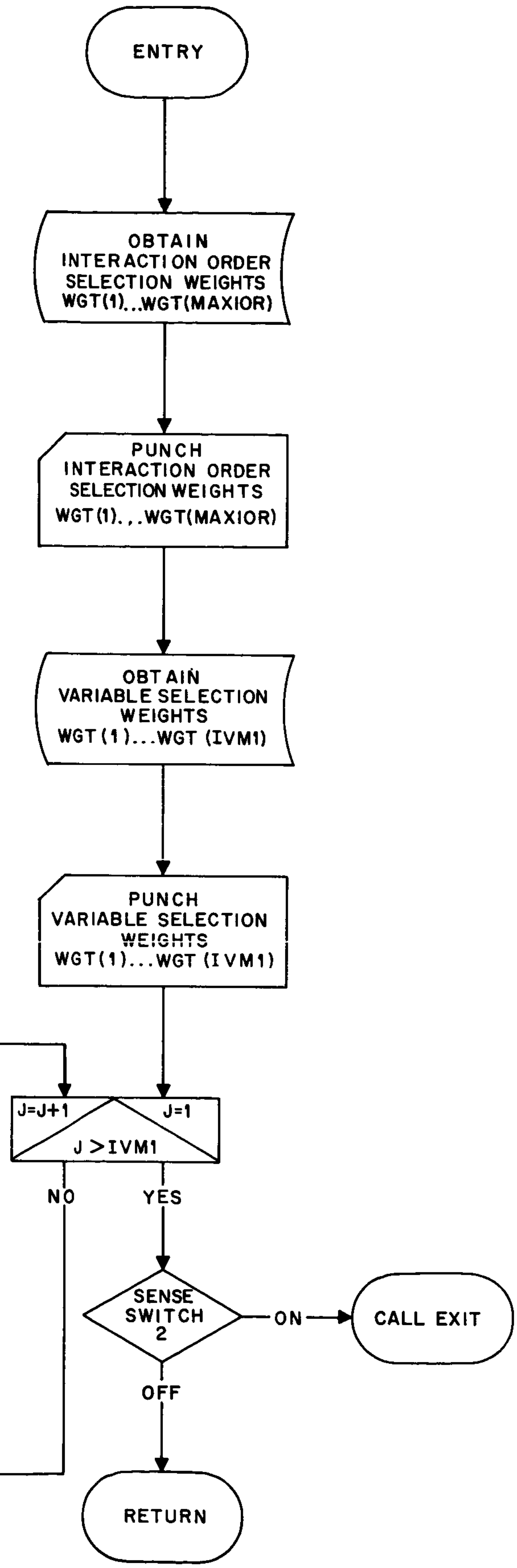


Figure E-26. SCORE-7 Subprogram -- Logic Flow Diagram

```
C
C                        *** DEFINITION OF VARIABLES ***
C
C     IDFN        DISC INDEX FOR LOCATING FUNCTION WEIGHTS
C     IDIOW       DISC INDEX FOR LOCATING INTERACTION ORDER WEIGHTS.
C     IDVARW      DISC INDEX FOR LOCATING VARIABLE WEIGHTS.
C     IVM1        NUMBER OF INPUT VARIABLES NOT INCLUDING THE DEPENDENT VARIABLE
C     MAXFUN      MAXIMUM NUMBER OF FUNCTIONS CONSIDERED
C     MAXIOR      MAXIMUM INTERACTION ORDER CONSIDERED.
C     NOTER       NUMBER OF TERMS CONSIDERED PER PASS INCLUDING THE DEPENDENT
C                 VARIABLE
C     WGT         TEMPORARY STORAGE ARRAY FOR SELECTION WEIGHTS
C
C                                  *** PROGRAM ***
C
C
C     SENSE SWITCH 1 ON - SELECTOR ARRAY WEIGHTS DUMPED
C
C     SENSE SWITCHES 1 AND 2 ON - DUMP AND CALL EXIT
C
      SUBROUTINE SCORE7(IDIOW,MAXIOR,IDVARW,IVM1,MAXFUN,NOTER,IDFN)
      DIMENSION IDFN(30),WGT(30),IV(30),IF(30)
  101 FORMAT(8F10.5)
      MD=IDIOW
      FETCH(MD)(WGT(I),I=1,MAXIOR)
      PUNCH 101,(WGT(I),I=1,MAXIOR)
      MD=IDVARW
      FETCH(MD)(WGT(I),I=1,IVM1)
      PUNCH 101,(WGT(I),I=1,IVM1)
      DO 1 J=1,IVM1
      MD=IDFN(J)
      FETCH(MD)(WGT(I),I=1,MAXFUN)
      PUNCH 101,J
    1 PUNCH 101,(WGT(I),I=1,MAXFUN)
    3 RETURN
      END
```

Figure E-27. SCORE-7 Subprogram -- FORTRAN Program Listing

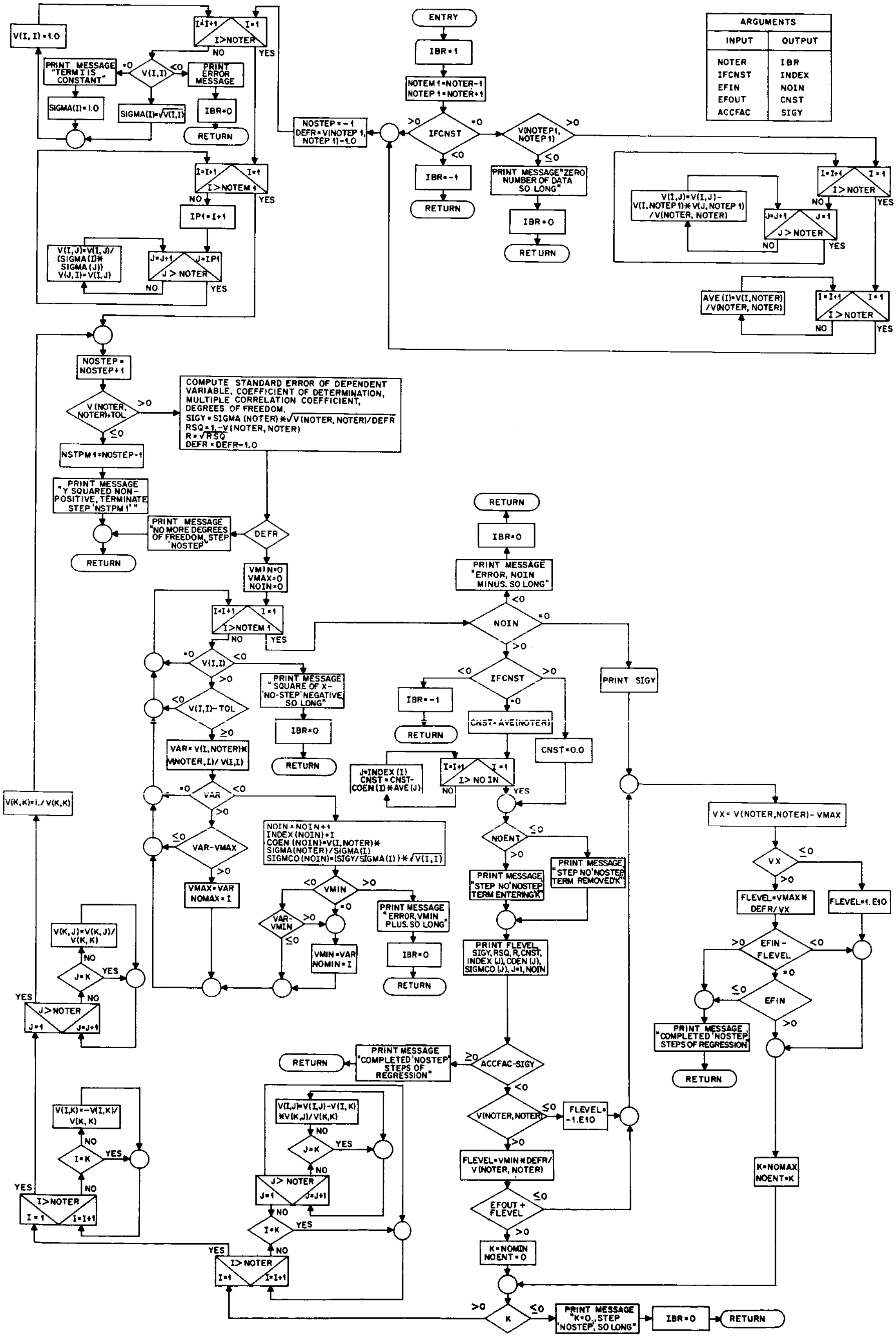


Figure E-28. SCORE-8 Subprogram -- Logic Flow Diagram

```
C
C                         *** DEFINITION OF VARIABLES ***
C
C     ACCFAC     STANDARD ERROR DESIRED FOR A SATISFACTORY REGRESSION RUN
C     AVE        AVERAGE VALUE ARRAY FOR AVAILABLE TERMS
C     CNST       CONSTANT TERM VALUE FOR REGRESSION EQUATION
C     COEN       REGRESSION EQUATION COEFFICIENT ARRAY
C     DEFR       NUMBER OF DEGREES OF FREEDOM
C     EFIN       CRITICAL F-LEVEL FOR ENTERING A TERM
C     EFOUT      CRITICAL F-LEVEL FOR REMOVING A TERM
C     FLEVEL     COMPUTED F-LEVEL FOR REMOVING LEAST SIGNIFICANT TERM OR
C                INSERTING MOST SIGNIFICANT TERM.
C     IBR        BRANCH INDICATOR
C     IFCNST     CONSTANT TERM INDICATOR. 1 IMPLIES NO CONSTANT TERM.
C                0 IMPLIES A CONSTANT TERM.
C     INDEX      ARRAY OF TERM NUMBERS IN THE REGRESSION EQUATION
C     NOIN       NUMBER OF TERMS IN THE REGRESSION EQUATION
C     NOMAX      INDEX NUMBER OF MOST SIGNIFICANT TERM NOT CURRENTLY IN THE
C                REGRESSION EQUATION
C     NOMIN      INDEX NUMBER OF LEAST SIGNIFICANT TERM CURRENTLY IN THE
C                REGRESSION EQUATION
C     NOSTEP     STEP NUMBER FOR CURRENT PASS
C     NOTEM1     NUMBER OF TERMS CONSIDERED PER PASS MINUS ONE
C     NOTEP1     NUMBER OF TERMS CONSIDERED PER PASS PLUS ONE
C     NOTER      NUMBER OF TERMS CONSIDERED PER PASS INCLUDING THE DEPENDENT
C                VARIABLE
C     NSTPM1     STEP NUMBER FOR CURRENT PASS MINUS ONE
C     R          MULTIPLE CORRELATION COEFFICIENT FOR THE CURRENT REGRESSION
C                EQUATION
C     RSQ        COEFFICIENT OF DETERMINATION FOR THE CURRENT REQRESSION
C                EQUATION
C     SIGMA      ARRAY CONTAINING THE SQUARE ROOT OF THE RESIDUAL SUM OF SQUARES
C                FOR EACH AVAILABLE TERM
C     SIGMCO     ARRAY CONTAINING THE STANDARD ERRORS OF THE COEFFICIENTS OF THE
C                REGRESSION EQUATION
C     SIGY       STANDARD ERROR FOR CURRENT REGRESSION EQUATION
C     V          REGRESSION MATRIX
C     VAR        ARRAY CONTAINING VARIANCE CONTRIBUTION OF EACH AVAILABLE TERM
C     VMAX       VALUE OF VARIANCE CONTRIBUTION FOR MOST SIGNIFICANT TERM NOT
C                CURRENTLY IN THE REGRESSION EQUATION
C     VMIN       VALUE OF VARIANCE CONTRIBUTION FOR LEAST SIGNIFICANT TERM
C                CURRENTLY IN THE REGRESSION EQUATION
C
C                              *** PROGRAM ***
C
      SUBROUTINE SCORE8(NOTER,IFCNST,IBR,EFIN,EFOUT,INDEX,NOIN,CNST,ACCF
     1AC,SIGY)
      DIMENSION V(31,31),AVE(30),SIGMA(30),COEN(30),INDEX(30),SIGMCO(30)
      COMMON V,COEN
      IBR=1
  565 NOTEM1=NOTER-1
  566 NOTEP1=NOTER+1
   20 FORMAT(4(1X,2HT(,I2,3H) =,E10.3))
```

Figure E-29. SCORE-8 Subprogram -- FORTRAN Program Listing (1 of 4)

```
   25 FORMAT( 3X,5HY    =,E10.3)
   35 FORMAT(3(1X,2HT(,I2,7H) VS T(,I2,2H)=,E10.3))
   40 FORMAT(3(1X,2HT(,I2,11H) VS    Y  =,E10.3))
   45 FORMAT (1X,15H  Y    VS    Y  =,E10.3)
C       CALCULATION OF RESIDUAL SUMS OF SQUARES AND CROSS PRODUCTS
  650 IF(IFCNST)900,651,780
  651 IF(V(NOTEP1,NOTEP1))652,652,655
  652 PRINT 654
  654 FORMAT(31H  ZERO NUMBER OF DATA. SO LONG.)
      GO TO 910
  655 DO 660 I=1,NOTER
  670 DO 660 J=1,NOTER
  660 V(I,J)=V(I,J)-(V(I,NOTEP1)*V(J,NOTEP1)/V(NOTEP1,NOTEP1))
  680 DO 690 I=1,NOTER
  690 AVE(I)=V(I,NOTEP1)/V(NOTEP1,NOTEP1)
  780 NOSTEP=-1
  782 DEFR=V(NOTEP1,NOTEP1)-1.0
  790 DO 800 I=1,NOTER
  791 IF(V(I,I))792,794,810
  792 PRINT 793,I
      GO TO 910
  793 FORMAT(31H ERROR RESIDUAL SQUARE VARIABLE,I4,31H IS NEGATIVE,PROBL
     1EM TERMINATED)
  794 PRINT 795,I
  795 FORMAT(//9X,5HTERM(,I2,13H) IS CONSTANT)
  796 SIGMA(I)=1.0
  797 GO TO 800
  810 SIGMA(I)=SQRTF(V(I,I))
  800 V(I,I)=1.0
  820 DO 830 I=1,NOTEM1
  840 IP1=I+1
  841 DO 830 J=IP1,NOTER
  850 V(I,J)=V(I,J)/(SIGMA(I)*SIGMA(J))
  830 V(J,I)=V(I,J)
 1000 NOSTEP=NOSTEP+1
 1001 IF(V(NOTER,NOTER)+.00001)1002,1002,1010
 1002 NSTPM1=NOSTEP-1
      PRINT 1004,NSTPM1
 1004 FORMAT(// ,38HY SQUARE NON-POSITIVE,TERMINATE STEP ,I5)
      GO TO 1381
 1010 SIGY=SIGMA(NOTER)*SQRTF(V(NOTER,NOTER)/DEFR)
 1011 RSQ=1.0-V(NOTER,NOTER)
      R=SQRTF(RSQ)
 1015 DEFR=DEFR-1.0
 1016 IF(DEFR)1018,1018,1020
 1018 PRINT 1019,NOSTEP
 1019 FORMAT(//,31H  NO MORE DEGREES FREEDOM STEP ,I5)
      GO TO 1381
 1020 VMIN=0.0
 1030 VMAX=0.0
 1035 NOIN=0
 1040 DO 1050 I=1,NOTEM1
 1041 IF(V(I,I))1043,1050,1060
```

Figure E-29. SCORE-8 Subprogram -- FORTRAN Program Listing (2 of 4)

```
1043 PRINT 1044,I,NOSTEP
1044 FORMAT(//,11H  SQUARE X-,I5,19H NEGATIVE. SO LONG ,I5,6H STEPS)
1045 GO TO 910
1060 IF(V(I,I)-.00001)1050,1080,1080
1080 VAR=V(I,NOTER)*V(NOTER,I)/V(I,I)
1090 IF(VAR)1100,1050,1110
1100 NOIN=NOIN+1
1120 INDEX(NOIN)=I
1130 COEN(NOIN)=V(I,NOTER)*SIGMA(NOTER)/SIGMA(I)
1140 SIGMCO(NOIN)=(SIGY/SIGMA(I))*SQRTF(V(I,I))
1150 IF(VMIN)1160,1170,904
 904 PRINT 906
 906 FORMAT(26H ERROR, VMIN PLUS. SO LONG)
     GO TO 910
1170 VMIN=VAR
1180 NOMIN=I
1190 GO TO 1050
1160 IF(VAR-VMIN)1050,1050,1170
1110 IF(VAR-VMAX)1050,1050,1210
1210 VMAX=VAR
1220 NOMAX=I
1050 CONTINUE
1230 IF(NOIN)903,1240,1245
 903 PRINT 907
 907 FORMAT(27H ERROR, NOIN MINUS. SO LONG)
     GO TO 910
1240 PRINT 65,SIGY
  65 FORMAT(//25H   STANDARD ERROR OF Y = ,E12.5//)
1260 GO TO 1350
1245 IF(IFCNST)900,1250,1246
1246 CNST=0.0
1247 GO TO 1300
1250 CNST=AVE(NOTER)
1270 DO 1280 I=1,NOIN
1290 J=INDEX(I)
1280 CNST=CNST-(COEN(I)*AVE(J))
1300 IF(NOENT)1311,1311,1313
1311 PRINT 91,NOSTEP,K
  91 FORMAT(//9H STEP NO ,I5,/3X,13HTERM REMOVED ,I8)
1312 GO TO 1314
1313 PRINT 92,NOSTEP,K
  92 FORMAT(//9H STEP NO ,I5,/3X,14HTERM ENTERING ,I8)
1314 PRINT 70,FLEVEL,SIGY,RSQ,R,CNST,(INDEX(J),COEN(J),SIGMCO(J),J=1,NO
    1IN)
  70 FORMAT(12H   F LEVEL =,E12.5/24H   STANDARD ERROR OF Y =,E12.5/27H
    1   COEFF OF DETERMINATION =,F12.8/26H   MULTIPLE CORLTN COEFF =,F1
    22.8//18H   CONSTANT TERM =,E12.5/58H      TERM     NO.         COEFFIC
    3IENT        STD ERR OF COEFF /(5X,2HT(,I2,1H),13X,E12.5,10X,E12.5))
1320 IF(ACCFAC-SIGY)104,1380,1380
 104 IF(V(NOTER,NOTER))101,101,102
 101 FLEVEL=-1.E10
     GO TO 1350
 102 FLEVEL=VMIN*DEFR/V(NOTER,NOTER)
```

Figure E-29. SCORE-8 Subprogram -- FORTRAN Program Listing (3 of 4)

```
1330 IF(EFOUT+FLEVEL)1350,1350,1340
1340 K=NOMIN
1345 NOENT=0
     GO TO 1391
1350 VX=V(NOTER,NOTER)-VMAX
     IF(VX)100,100,1360
 100 FLEVEL=1.E10
     GO TO 1370
1360 FLEVEL=VMAX*DEFR/VX
     IF(EFIN-FLEVEL)1370,1361,1380
1361 IF(EFIN)1380,1380,1370
1370 K=NOMAX
1390 NOENT=K
1391 IF(K) 1392,1392,1400
1392 PRINT 1395,NOSTEP
1395 FORMAT(12H  K=0. STEP ,I6,8H SO LONG)
1394 GO TO 910
1400 DO 1410 I=1,NOTER
1420 IF(I-K)1430,1410,1430
1430 DO 1440 J=1,NOTER
1450 IF(J-K)1460,1440,1460
1460 V(I,J)=V(I,J)-(V(I,K)*V(K,J)/V(K,K))
1440 CONTINUE
1410 CONTINUE
1470 DO 1480 I=1,NOTER
1490 IF(I-K)1500,1480,1500
1500 V(I,K)=-V(I,K)/V(K,K)
1480 CONTINUE
1510 DO 1520 J=1,NOTER
1530 IF(J-K)1540,1520,1540
1540 V(K,J)=V(K,J)/V(K,K)
1520 CONTINUE
1550 V(K,K)=1.0/V(K,K)
1560 GO TO 1000
1380 PRINT 75,NOSTEP
  75 FORMAT(11H COMPLETED ,I5,20H STEPS OF REGRESSION)
1381 RETURN
 900 IBR=-1
     RETURN
 910 IBR=0
     RETURN
     END
```

Figure E-29. SCORE-8 Subprogram -- FORTRAN Program Listing (4 of 4)

4 SCORE APPLICATION TO A SAMPLE PROBLEM

In order to demonstrate the effectiveness of the SCORE program in constructing an equation from a set of data, the following sample problem was run.

Starting with the equation

$$y = 2X_1^2 - 4X_1 - 3X_2^2 + 12X_2 - 10 + E$$

a set of twenty-five data points was generated for various values of X_1 and X_2. A noise factor (E in the above equation) was added to more closely simulate a set of laboratory measurements. This noise factor was selected at random from a normal distribution having a mean of zero and a standard deviation of 0.1.

Data Transmittal Form I as filled out for this problem is shown in Figure E-30. Counting the blank cards separating the comments, a total of nine cards are used as comment cards. Therefore, a 9 is entered on Card 1 as shown. Following this card are the comment cards.

Data Transmittal Form II is shown in Figure E-31. Assuming a maximum allowable probability of committing the error of entering a variable into the regression equation when it should not be entered to be .05, ($p = .05$) and entering the F-level chart given as Table E-II on Page E-31 for 25 variables ($n = 25$), gives a value of F-level for entering approximately equal to 4. This value is inserted as F-LEVEL IN on Control Card 1.

Assuming a maximum allowable probability of committing the error of removing a variable from the regression equation when it should not have been removed to be .25 ($p = .25$), and entering the F-level chart, gives a value of F-level for removing (using interpolation) approximately equal to 1. This value is inserted as F-LEVEL OUT on Control Card 1. Assuming a desired standard deviation for the errors in the regression equation to be .01, this value is entered as STD. DEVIATION on Control Card 1. Since a constant term is expected for the regression equation, the CONSTANT indicator is set to 0. The number of variables is two independent and one dependent for a total of 3. Therefore, NO. VARIABLES is set to 3. There are 25 data points for this particular problem. Therefore, NO. DATA is set to 25.

LEAR SIEGLER, INC. INSTRUMENT DIVISION	SCORE DTF I	REGRESSION ANALYSIS	SHEET 1 OF 3

CARD 1

NO. COMMENTS
9

COMMENT CARDS

```
                    SCORE SAMPLE PROBLEM
             REGRESSION ANALYSIS FOR THE EQUATION
            Y=2.*X(1)**2-4.*X(1)-3.*X(2)**2+12.*X(2)-10.
        WITH WHITE NOISE HAVING A MEAN OF 0. AND A SIGMA OF 0.1
                      DATE 8/25/65
```

Figure E-30. SCORE Data Transmittal Form I, Sample Problem Inserted

ISi	**LEAR SIEGLER, INC.** INSTRUMENT DIVISION	SCORE DTF II	**REGRESSION ANALYSIS**	SHEET 2 OF 3

CONTROL CARD 1

F - LEVEL IN (1–10)	F - LEVEL OUT (11–20)	STD. DEVIATION (21–30)	CONSTANT (31–40)	NO. VARIABLES (41–50)	NO. DATA (51–60)	61–80
4.	1.	.01	0.	3.	25.	

CONTROL CARD 2

NO. TERMS (1–10)	INTERACTION ORDER (11–20)	INTERACTION WEIGHT (21–30)	VARIABLE WEIGHT (31–40)	NO. FUNCTIONS (41–50)	FUNCTION WEIGHT (51–60)	61–80
10.	2.	1.	1.	6.	1.	

INPUT VARIABLES

IDENTIFICATION (1–10)	DEPENDENT VAR. (11–20)	INDEPENDENT VARIABLES (Fill only those used) 21–30	31–40	41–50	51–60	61–70	71–80
1.	-.52151 1	.4 2	1.4 3	4	5	6	
	7	8	9	10	11	12	
	13	14	15	16	17	18	
	19	20	21	22	23	24	
	25	26	27	28			
2.	.73626	.4	1.8				
3.	.78340	.4	2.2				
4.	-.24243	.4	2.6				
5.	-2.17773	.4	3.0				
6.	-.93969	.8	1.4				
7.	-.02146	.8	1.8				
8.	-.09224	.8	2.2				
9.	-.88535	.8	2.6				
10.	-3.01356	.8	3.0				
11.	-1.16559	1.2	1.4				
12.	-.11597	1.2	1.8				

Figure E-31. SCORE Data Transmittal Form II, Sample Problem Inserted

A reasonable number of terms to be considered per pass for a three variable problem is 10. Therefore, NO. TERMS on Control Card 2 is set to 10.

The maximum desired interaction order must be limited to 2 since there are only 2 independent variables. Also products of X_1 and X_2 were to be considered in order to complicate the problem. Therefore, INTERACTION ORDER is set to 2. Since no a prior information was to be given the program, all initial selection weights were set equal. Therefore, INTERACTION WEIGHT, VARIABLE WEIGHT, and FUNCTION WEIGHT indicators are all set to 1. The number of functions to be considered for each independent variable was set to 6. This allows the powers 1, -1, 2, -2, 1/2 and -1/2 to be applied to the independent variables. Therefore, NO. FUNCTIONS was set to 6. (See Table E-I for a definition of the allowable functions).

The next set of cards to be prepared contains the actual data point values for the two independent variables and the dependent variable along with an identification number for each data point. These entries are shown on Data Transmittal Forms II and III. (Figures E-31 and E-32.) Submitting these data cards to the SCORE program results in the computer output sheets as shown in Figure E-33.

Initial output consists of the comment cards prepared as inputs, the control values of F-level for entering and F-level for removing, and the desired value for the standard error of the regression equation. Following this is a print-out of the input data points.

Pass number 1 is then begun. The ten available terms as randomly constructed for this pass are then listed. Following this is a step-by-step record of the actual equation construction along with other pertinent information about the quality of the fit at each step. After Step 4 of the first pass, there are no more terms available which meet the requirements for entry into the equation and there are no terms in the equation which do not meet the requirements for remaining.

A listing of the predicted values for the dependent variable using the regression equation constructed during Pass 1 along with the actual input values is then given for each data point.

Pass number 2 is then begun. Those terms which successfully entered the regression equation during Pass 1 (namely T(1), T(8), T(9), and T(10)) are now included in the set of 10 available terms for Pass 2. The remaining six terms are again randomly constructed, this time taking advantage of what was learned during Pass 1 (see previous text for a discussion of the learning process).

LEAR SIEGLER, INC. INSTRUMENT DIVISION	SCORE DTF III	REGRESSION ANALYSIS	SHEET 3 OF 3

INPUT VARIABLES (Continued)

IDENTIFICATION (1–10)	DEPENDENT VAR. (11–20)	INDEPENDENT VARIABLES (Fill only those used) 21–30	31–40	41–50	51–60	61–70	71–80
13.	-.10048	1.2	2.2				
14.	-1.00096	1.2	2.6				
15.	-2.84658	1.2	3.0				
16.	-.42282	1.6	1.4				
17.	.41672	1.6	1.8				
18.	.54709	1.6	2.2				
19.	-.40347	1.6	2.6				
20.	-2.24687	1.6	3.0				
21.	1.09706	2.0	1.4				
22.	1.90312	2.0	1.8				
23.	1.89336	2.0	2.2				
24.	.81413	2.0	2.6				
25.	-.94812	2.0	3.0				

Figure E-32. SCORE Data Transmittal Form III, Sample Problem Inserted

```
                    SCORE SAMPLE PROGRAM

            REGRESSION ANALYSIS FOR THE EQUATION

       Y=2.*X(1)**2-4.*X(1)-3.*X(2)**2+12.*X(2)-10.

  WITH WHITE NOISE HAVING A MEAN OF 0. AND A SIGMA OF 0.1

                       DATE  8/25/65

F LEVEL TO ENTER VARIABLE =  4.000000

F LEVEL TO REMOVE VARIABLE =  1.000000

DESIRED VALUE OF STANDARD ERROR =   .010000


DATA SET     1.
X( 1) = 4.000E-01 X( 2) = 1.400E-00
  Y   =-5.215E-01

DATA SET     2.
X( 1) = 4.000E-01 X( 2) = 1.800E-00
  Y   = 7.362E-01

DATA SET     3.
X( 1) = 4.000E-01 X( 2) = 2.200E-00
  Y   = 7.834E-01

DATA SET     4.
X( 1) = 4.000E-01 X( 2) = 2.600E-00
  Y   =-2.424E-01

DATA SET     5.
X( 1) = 4.000E-01 X( 2) = 3.000E-00
  Y   =-2.177E-00

DATA SET     6.
X( 1) = 8.000E-01 X( 2) = 1.400E-00
  Y   =-9.396E-01

DATA SET     7.
X( 1) = 8.000E-01 X( 2) = 1.800E-00
  Y   =-2.146E-02

DATA SET     8.
X( 1) = 8.000E-01 X( 2) = 2.200E-00
  Y   =-9.224E-02
```

Figure E-33.

Sample Problem Computer Output Sheets (1 of 9)

```
DATA SET    9.
X( 1) = 8.000E-01 X( 2) = 2.600E-00
  Y    =-8.853E-01

DATA SET    10.
X( 1) = 8.000E-01 X( 2) = 3.000E-00
  Y    =-3.013E-00

DATA SET    11.
X( 1) = 1.200E-00 X( 2) = 1.400E-00
  Y    =-1.165E-00

DATA SET    12.
X( 1) = 1.200E-00 X( 2) = 1.800E-00
  Y    =-1.159E-01

DATA SET    13.
X( 1) = 1.200E-00 X( 2) = 2.200E-00
  Y    =-1.004E-01

DATA SET    14.
X( 1) = 1.200E-00 X( 2) = 2.600E-00
  Y    =-1.000E-00

DATA SET    15.
X( 1) = 1.200E-00 X( 2) = 3.000E-00
  Y    =-2.846E-00

DATA SET    16.
X( 1) = 1.600E-00 X( 2) = 1.400E-00
  Y    =-4.228E-01

DATA SET    17.
X( 1) = 1.600E-00 X( 2) = 1.800E-00
  Y    = 4.167E-01

DATA SET    18.
X( 1) = 1.600E-00 X( 2) = 2.200E-00
  Y    = 5.470E-01

DATA SET    19.
X( 1) = 1.600E-00 X( 2) = 2.600E-00
  Y    =-4.034E-01

DATA SET    20.
X( 1) = 1.600E-00 X( 2) = 3.000E-00
  Y    =-2.246E-00
```

Figure E-33.

Sample Problem Computer Output Sheets (2 of 9)

```
DATA SET   21.
X( 1) = 2.000E-00 X( 2) = 1.400E-00
  Y   = 1.097E-00

DATA SET   22.
X( 1) = 2.000E-00 X( 2) = 1.800E-00
  Y   = 1.903E-00

DATA SET   23.
X( 1) = 2.000E-00 X( 2) = 2.200E-00
  Y   = 1.893E-00

DATA SET   24.
X( 1) = 2.000E-00 X( 2) = 2.600E-00
  Y   = 8.141E-01

DATA SET   25.
X( 1) = 2.000E-00 X( 2) = 3.000E-00
  Y   =-9.481E-01
```

Figure E-33.

Sample Problem Computer Output Sheets (3 of 9)

```
                              PASS NUMBER    1

TERM DESCRIPTIONS

TERM( 1) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 1)**(-2.000)

TERM( 2) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**(-1.000)

TERM( 3) = INTERACTION OF ORDER  2 WITH.
           COMPONENT( 1) = X( 1)**(-1.000)
           COMPONENT( 2) = X( 2)**(-2.000)

TERM( 4) = INTERACTION OF ORDER  2 WITH.
           COMPONENT( 1) = X( 1)**( -.500)
           COMPONENT( 2) = X( 2)**( 1.000)

TERM( 5) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**( -.500)

TERM( 6) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**(-2.000)

TERM( 7) = INTERACTION OF ORDER  2 WITH.
           COMPONENT( 1) = X( 1)**( 1.000)
           COMPONENT( 2) = X( 2)**(-2.000)

TERM( 8) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**( 2.000)

TERM( 9) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 1)**( 2.000)

TERM(10) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**( 1.000)

  STANDARD ERROR OF Y =  1.29263E-00
```

Figure E-33.

Sample Problem Computer Output Sheets (4 of 9)

```
STEP NO      1
  TERM ENTERING          8
  F LEVEL = 1.18408E+01
  STANDARD ERROR OF Y = 1.07284E-00
  COEFF OF DETERMINATION =    .33985582
  MULTIPLE CORLTN COEFF =    .58297154

  CONSTANT TERM = 1.16374E-00
    TERM    NO.          COEFFICIENT          STD ERR OF COEFF
    T( 8)                -2.94939E-01             8.57118E-02


STEP NO      2
  TERM ENTERING          10
  F LEVEL = 2.21918E+01
  STANDARD ERROR OF Y = 7.73979E-01
  COEFF OF DETERMINATION =    .67136098
  MULTIPLE CORLTN COEFF =    .81936620

  CONSTANT TERM =-1.13800E+01
    TERM    NO.          COEFFICIENT          STD ERR OF COEFF
    T( 8)                -3.00300E-00             5.78176E-01
    T(10)                 1.20533E+01             2.55865E-00


STEP NO      3
  TERM ENTERING          9
  F LEVEL = 2.47186E+01
  STANDARD ERROR OF Y = 5.36900E-01
  COEFF OF DETERMINATION =    .84904594
  MULTIPLE CORLTN COEFF =    .92143689

  CONSTANT TERM =-1.20590E+01
    TERM    NO.          COEFFICIENT          STD ERR OF COEFF
    T( 8)                -3.00300E-00             4.01074E-01
    T( 9)                 3.85803E-01             7.75984E-02
    T(10)                 1.20533E+01             1.77490E-00


STEP NO      4
  TERM ENTERING          1
  F LEVEL = 1.05332E+02
  STANDARD ERROR OF Y = 2.19772E-01
  COEFF OF DETERMINATION =    .97591133
  MULTIPLE CORLTN COEFF =    .98788224

  CONSTANT TERM =-1.31588E+01
    TERM    NO.          COEFFICIENT          STD ERR OF COEFF
    T( 1)                 2.84199E-01             2.76913E-02
    T( 8)                -3.00300E-00             1.64173E-01
    T( 9)                 7.15279E-01             4.51611E-02
    T(10)                 1.20533E+01             7.26531E-01
COMPLETED     4 STEPS OF REGRESSION
```

Figure E-33. Sample Problem Computer Output Sheets (5 of 9)

PREDICTED VS ACTUAL RESULTS

RUN NO.	ACTUAL	PREDICTED	DEVIATION	PERCENT DEV.
1.	-5.21510E-01	-2.79360E-01	2.42149E-01	4.64323E+01
2.	7.36260E-01	6.98135E-01	-3.81248E-02	-5.17817E-00
3.	7.83400E-01	7.14668E-01	-6.87312E-02	-8.77345E-00
4.	-2.42430E-01	-2.29759E-01	1.26703E-02	5.22640E-00
5.	-2.17773E-00	-2.13515E-00	4.25799E-02	1.95524E-00
6.	-9.39690E-01	-1.26821E-00	-3.28523E-01	-3.49608E+01
7.	-2.14600E-02	-2.90717E-01	-2.69257E-01	-1.25469E+03
8.	-9.22400E-02	-2.74184E-01	-1.81944E-01	-1.97250E+02
9.	-8.85350E-01	-1.21861E-00	-3.33262E-01	-3.76419E+01
10.	-3.01356E-00	-3.12400E-00	-1.10443E-01	-3.66487E-00
11.	-1.16559E-00	-9.42691E-01	2.22898E-01	1.91232E+01
12.	-1.15970E-01	3.48045E-02	1.50774E-01	1.30011E+02
13.	-1.00480E-01	5.13381E-02	1.51818E-01	1.51092E+02
14.	-1.00096E-00	-8.93090E-01	1.07869E-01	1.07766E+01
15.	-2.84658E-00	-2.79848E-00	4.80993E-02	1.68972E-00
16.	-4.22820E-01	-2.27923E-01	1.94896E-01	4.60945E+01
17.	4.16720E-01	7.49572E-01	3.32852E-01	7.98743E+01
18.	5.47090E-01	7.66106E-01	2.19016E-01	4.00329E+01
19.	-4.03470E-01	-1.78322E-01	2.25147E-01	5.58028E+01
20.	-2.24687E-00	-2.08371E-00	1.63157E-01	7.26153E-00
21.	1.09706E-00	7.62114E-01	-3.34945E-01	-3.05312E+01
22.	1.90312E-00	1.73960E-00	-1.63510E-01	-8.59169E-00
23.	1.89336E-00	1.75614E-00	-1.37216E-01	-7.24725E-00
24.	8.14130E-01	8.11714E-01	-2.41512E-03	-2.96650E-01
25.	-9.48120E-01	-1.09367E-00	-1.45555E-01	-1.53520E+01

Figure E-33.

Sample Problem Computer Output Sheets (6 of 9)

```
                              PASS NUMBER   2

TERM DESCRIPTIONS

TERM( 1) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 1)**(-2.000)

TERM( 2) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**( 2.000)

TERM( 3) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 1)**( 2.000)

TERM( 4) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**( 1.000)

TERM( 5) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 1)**( 1.000)

TERM( 6) = INTERACTION OF ORDER  2 WITH.
           COMPONENT( 1) = X( 1)**( 2.000)
           COMPONENT( 2) = X( 2)**( 1.000)

TERM( 7) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**(  .500)

TERM( 8) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**(-1.000)

TERM( 9) = INTERACTION OF ORDER  1 WITH.
           COMPONENT( 1) = X( 2)**( -.500)

TERM(10) = INTERACTION OF ORDER  2 WITH.
           COMPONENT( 1) = X( 1)**( 2.000)
           COMPONENT( 2) = X( 2)**(  .500)

  STANDARD ERROR OF Y =   1.29263E-00
```

Figure E-33.

Sample Problem Computer Output Sheets (7 of 9)

```
STEP NO      1
  TERM ENTERING            2
  F LEVEL = 1.18408E+01
  STANDARD ERROR OF Y = 1.07284E-00
  COEFF OF DETERMINATION =    .33985582
  MULTIPLE CORLTN COEFF =    .58297154

  CONSTANT TERM = 1.16374E-00
    TERM    NO.          COEFFICIENT          STD ERR OF COEFF
    T( 2)                -2.94939E-01              8.57118E-02


STEP NO      2
  TERM ENTERING            4
  F LEVEL = 2.21918E+01
  STANDARD ERROR OF Y = 7.73979E-01
  COEFF OF DETERMINATION =    .67136098
  MULTIPLE CORLTN COEFF =    .81936620

  CONSTANT TERM =-1.13800E+01
    TERM    NO.          COEFFICIENT          STD ERR OF COEFF
    T( 2)                -3.00300E-00              5.78176E-01
    T( 4)                 1.20533E+01              2.55865E-00


STEP NO      3
  TERM ENTERING            3
  F LEVEL = 2.47186E+01
  STANDARD ERROR OF Y = 5.36900E-01
  COEFF OF DETERMINATION =    .84904594
  MULTIPLE CORLTN COEFF =    .92143689

  CONSTANT TERM =-1.20590E+01
    TERM    NO.          COEFFICIENT          STD ERR OF COEFF
    T( 2)                -3.00300E-00              4.01074E-01
    T( 3)                 3.85803E-01              7.75984E-02
    T( 4)                 1.20533E+01              1.77490E-00

STEP NO      4
  TERM ENTERING            5
  F LEVEL = 6.13745E+02
  STANDARD ERROR OF Y = 9.77340E-02
  COEFF OF DETERMINATION =    .99523613
  MULTIPLE CORLTN COEFF =    .99761522

  CONSTANT TERM =-9.87280E-00
    TERM    NO.          COEFFICIENT          STD ERR OF COEFF
    T( 2)                -3.00300E-00              7.30091E-02
    T( 3)                 2.16034E-00              7.30091E-02
    T( 4)                 1.20533E+01              3.23093E-01
    T( 5)                -4.42453E-00              1.78596E-01
COMPLETED       4 STEPS OF REGRESSION
```

Figure E-33. Sample Problem Computer Output Sheets (8 of 9)

PREDICTED VS ACTUAL RESULTS

RUN NO.	ACTUAL	PREDICTED	DEVIATION	PERCENT DEV.
1.	-5.21510E-01	-3.08146E-01	2.13363E-01	4.09126E+01
2.	7.36260E-01	6.69349E-01	-6.69108E-02	-9.08793E-00
3.	7.83400E-01	6.85882E-01	-9.75172E-02	-1.24479E+01
4.	-2.42430E-01	-2.58545E-01	-1.61156E-02	-6.64755E-00
5.	-2.17773E-00	-2.16393E-00	1.37939E-02	6.33409E-01
6.	-9.39690E-01	-1.04099E-00	-1.01301E-01	-1.07803E+01
7.	-2.14600E-02	-6.34961E-02	-4.20361E-02	-1.95881E+02
8.	-9.22400E-02	-4.69625E-02	4.52774E-02	4.90865E+01
9.	-8.85350E-01	-9.91390E-01	-1.06040E-01	-1.19772E+01
10.	-3.01356E-00	-2.89678E-00	1.16778E-01	3.87510E-00
11.	-1.16559E-00	-1.08252E-00	8.30640E-02	7.12635E-00
12.	-1.15970E-01	-1.05030E-01	1.09396E-02	9.43317E-00
13.	-1.00480E-01	-8.84967E-02	1.19832E-02	1.19260E+01
14.	-1.00096E-00	-1.03292E-00	-3.19651E-02	-3.19344E-00
15.	-2.84658E-00	-2.93831E-00	-9.17355E-02	-3.22265E-00
16.	-4.22820E-01	-4.32748E-01	-9.92897E-03	-2.34827E-00
17.	4.16720E-01	5.44746E-01	1.28026E-01	3.07224E+01
18.	5.47090E-01	5.61280E-01	1.41902E-02	2.59376E-00
19.	-4.03470E-01	-3.83148E-01	2.03218E-02	5.03676E-00
20.	-2.24687E-00	-2.28853E-00	-4.16685E-02	-1.85451E-00
21.	1.09706E-00	9.08339E-01	-1.88720E-01	-1.72024E+01
22.	1.90312E-00	1.88583E-00	-1.72852E-02	-9.08258E-01
23.	1.89336E-00	1.90236E-00	9.00834E-03	4.75786E-01
24.	8.14130E-01	9.57939E-01	1.43809E-01	1.76642E+01
25.	-9.48120E-01	-9.47450E-01	6.69542E-04	7.06179E-02

Figure E-33.

Sample Problem Computer Output Sheets (9 of 9)

Now a new regression equation is constructed. This time the terms of the original equation used to construct the input data are all included in the 10 available terms (namely T(2) which is the X_2^2 term, T(3) which is the X_1^2 term, T(4) which is the X_2 term, and T(5) which is the X_1 term). These four terms are selected by the program and Pass 2 is terminated. Note the value of standard error for the dependent variable after Step 4 to be .0977. This is to be expected, since the noise superimposed on the input data has a standard deviation of .1.

The final equation for Pass 2 is

$$y = 2.16 X_1^2 - 4.424 X_1 - 3.003 X_2^2 + 12.05 X_2 - 9.8728$$

which agrees very well with the initial equation from which the data was generated.

REFERENCES

1. Dallemand, James E., Stepwise Regression Program on the IBM 704, General Motors Corporation, 1958.

2. Spiegel, Murray B., Ph.D., Theory and Problems of Statistics, Schaum Publishing Co., 1961.

3. Ralston and Wilf, Mathematical Methods for Digital Computers, John Wiley and Sons, Inc., 1960.

APPENDIX F

OPTIMIZATION

(Generalized Random Extremum Analysis Technique)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1 INTRODUCTION

In recent years considerable time and money have been expended on the development of techniques for system optimization. This interest can generally be attributed to the complexity of modern systems, to the high cost of these systems, and to the extreme performance and reliability characteristics demanded of these systems. The interest in system optimization at LSI led eventually to the development of the PRESTO* concept.

The role of optimization in the PRESTO methodology is that of determining a combination of the system parameters which yield an optimum system. Mathematically, this is equivalent to maximizing or minimizing a function of the system parameters. Unfortunately, the functions which describe modern complex systems, such as the Launch Vehicle System, are not well behaved. That is, the functions are either highly non-linear, non-linearly constrained, or they or their derivatives do not satisfy certain conditions of continuity and differentiability. For this class of problems, a closed-form analytical solution cannot be obtained. In these cases the use of a high-speed digital computer is required in the determination of the optimal solution.

The technique developed to determine the optimal system configuration within which the constraints on the system parameters are satisfied is a Generalized Random Extremum Analysis Technique. GREAT can perhaps best be described as a combination of the direct search and gradient techniques to which has been added a learning capability.

*Performance, Reliability, and Economics Simulation Techniques for Optimization.

2 FUNCTION

When considered within the framework of the PRESTO concept, the objective function (the function to be optimized) can be either one or a combination of three models: the Economics model, the Reliability model, and the Performance model. A possible strategy might be to minimize the cost of a system while maintaining a reliability limit and various performance specifications. Another possibility might be to maximize the system reliability subject to a cost constraint. However, for the sake of generality the reference will be to only an objective function and to constraining functions.

2.1 OBJECTIVE AND CONSTRAINING FUNCTIONS

The objective function may be thought of as the functional relationship of the system parameters which describe a particular system characteristic presented for optimization. The constraining functions are functional relationships of the constituent system parameters by which these system parameters are restricted.

The dependent variable Y is a function f of N independent variables X_1, X_2, X_3, ... , X_N. The function is defined in a domain given by

$$X_{n(min)} \leq X_n \leq X_{n(max)}$$

$$n = 1, 2, 3, \ldots, N.$$

The quantities $X_{n(min)}$ and $X_{n(max)}$ are the minimum and maximum values, respectively, of the N independent variables.

Each constraint G_j is a function g_j of all, or part of, the same N independent variables X_1, X_2, X_3, ..., X_N ($j = 1, 2, \ldots p$; where p is the number of constraining functions). These functions are defined in the same domain as given above.

The objective function is restricted only in that it must exist everywhere in the N space. The constraining functions are subjected to an additional restriction in that these functions must be inequality functions of the form

$$G_j = g_j (X_1, X_2, X_3, \ldots, X_N) \geq 0.$$

The intervals

$$\left[X_{n(min)},\ X_{n(max)}\right]$$

of each of the N independent variables are divided into M_n -1 equal sub-intervals

$$\left[X_{n(1)},\ X_{n(2)}\right),\ \left[X_{n(2)},\ X_{n(3)}\right),\ \ldots,$$
$$\left[X_{n(M_n-2)},\ X_{n(M_n-1)}\right),\ \left[X_{n(M_n-1)},\ X_{n(M_n)}\right]$$

where

$$X_{n(1)} = X_{n(min)}$$

and

$$X_{n(M_n)} = X_{n(max)}.$$

Define the interval size

$$D X_n = \frac{X_{n(max)} - X_{n(min)}}{M_n - 1}.$$

3 GREAT METHODOLOGY

The Generalized Random Extremum Analysis Technique (GREAT) is a procedure for locating the extrema of an objective function by a systematic evaluation of the objective and constraining functions. (See Figure F-1.) The search for the extremum originates at a probabilistic starting point in the N dimensional feasible space. The search is initiated and continued in the direction of steepest descent until a local extremum* on the search direction has been located or until a boundary or a constraint is encountered. If the conditions for optimality are not satisfied at the local extremum (the boundary or constraint point), a new direction of search is computed by the steepest descent or auxiliary methods. The procedure is iterated until a relative extremum† has been found. At each iteration a learning scheme is utilized which affords a means of profiting from the previous investigations. After a suitable number of investigations, the global extremum† is selected from the group of relative extrema.

3.1 PROBABILISTIC STARTING POINT SELECTION

A Starting Point is selected by a probabilistic selection procedure which, by the multiplicity of these starting points, tends to assure a complete investigation of the objective function. The degree of confidence which can be assumed in the obtained solution being the global extremum is proportional to the completeness of this investigation.

3.1.1 Weighting Function

The task of generating the probabilistic points is accomplished by the utilization of a Weighting Density Function and its integral, the Weighting Distribution Function.

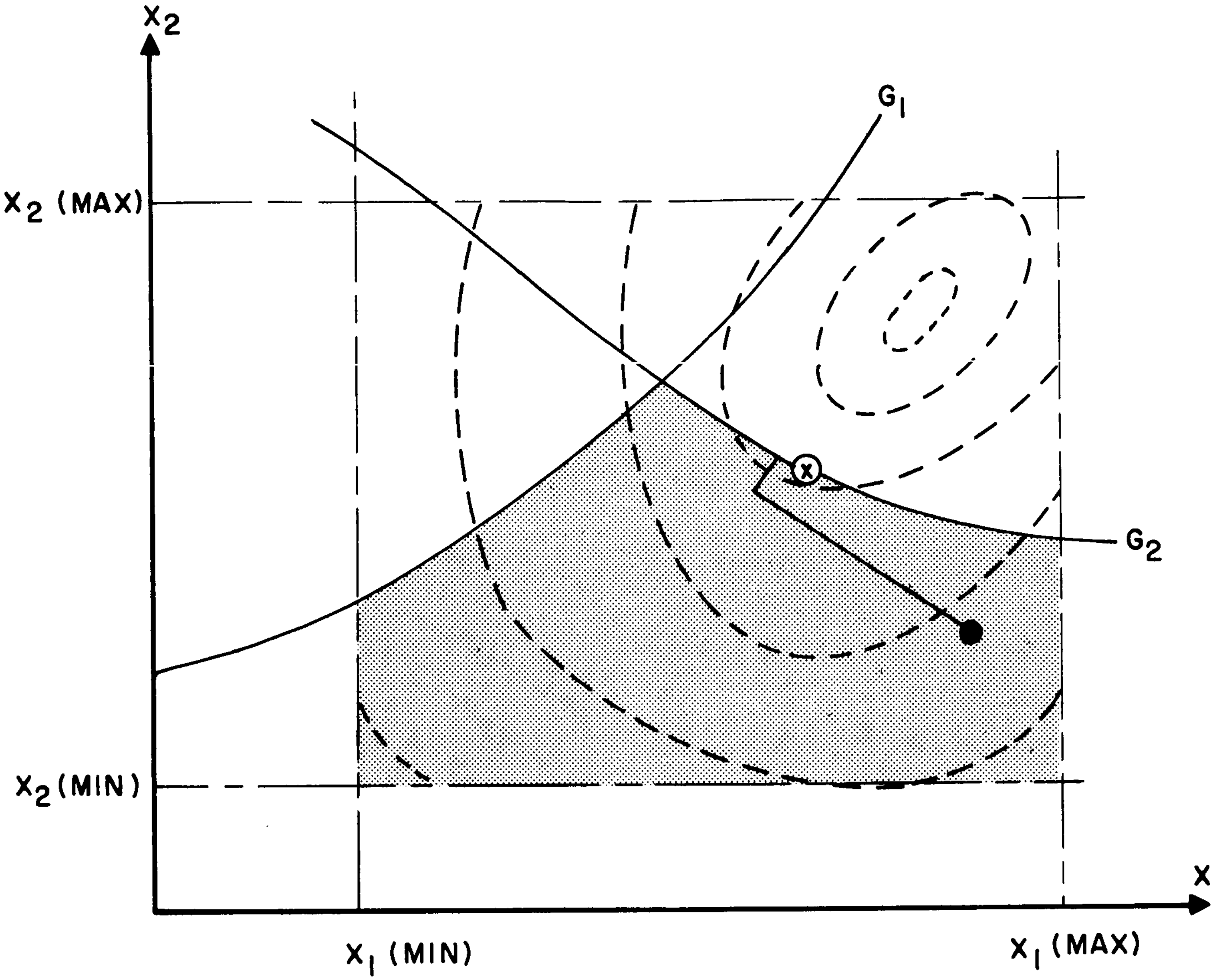
The weighting density function (Figure F-2) is defined by

$$W(X_n) = \begin{cases} 0 \text{ for } X_n < X_{n(min)} \\ W_{n(m)} \text{ for } X_{n(m)} \leq X_n \leq X_{n(m+1)} \\ 0 \text{ for } X_n > X_{n(max)} \end{cases}$$

*The local extremum is the extremum on a particular search direction.

†A relative extremum is that point from which no further optimization can be made.

†The global extremum is merely the most extreme point in a region.



BOUNDARIES

CONSTRAINTS (G_1, G_2)

OBJECTIVE FUNCTION EQUIPOTENTIAL LINES

SHADED AREA — FEASIBLE REGION

● — FEASIBLE STARTING POINT

SEARCH DIRECTION

⊗ — OPTIMUM

Figure F-1. Graphical Definition of Terminology

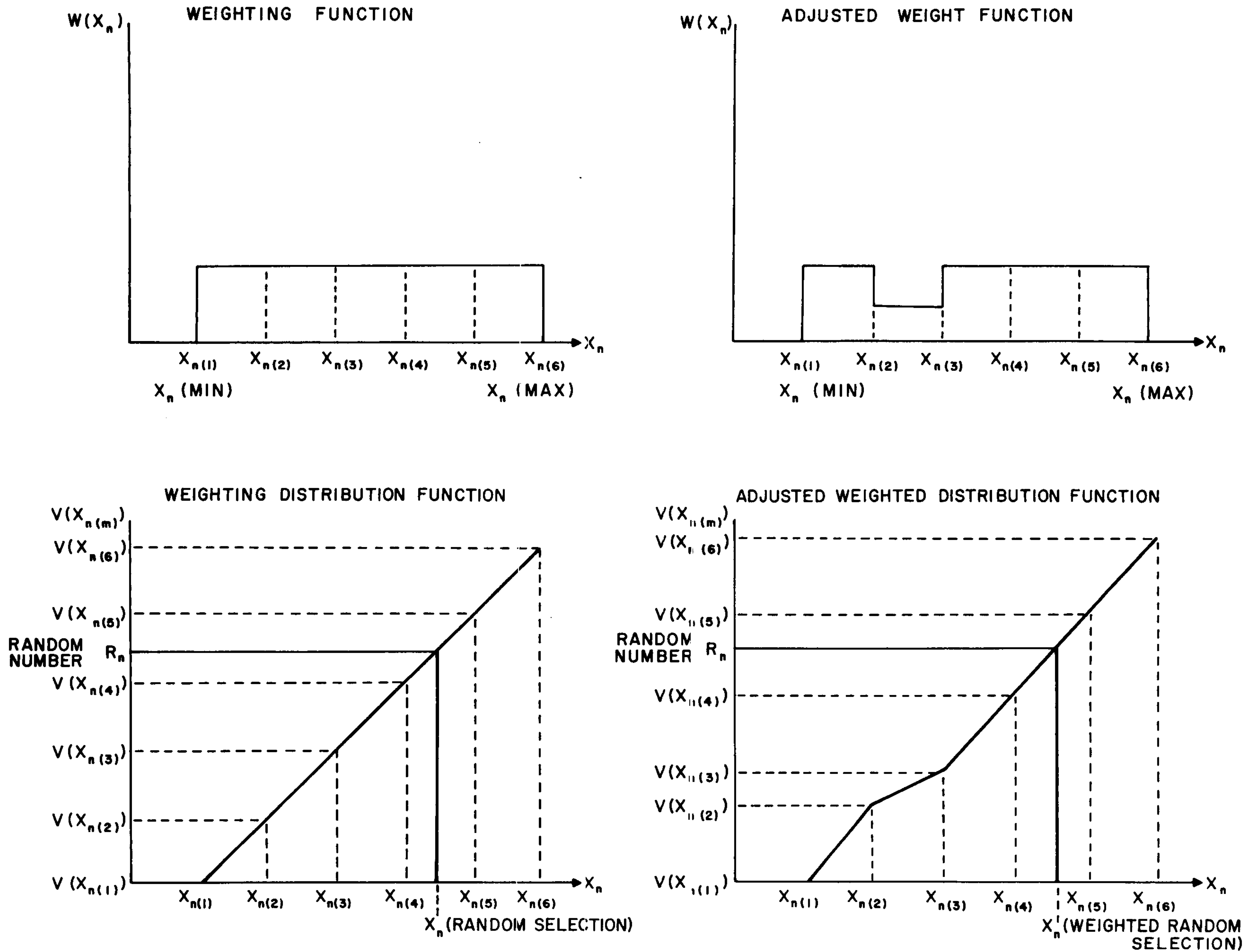


Figure F-2. Probabilistic Selection Procedure where $\left[X_{n(m)}, X_{n(m+1)}\right]$ is the m^{th} subinterval and $W_{n(m)}$ is some constant in the m^{th} subinterval and where

$$\int_{-\infty}^{+\infty} W(X_n)\, d\, X_n = \int_{X_{n(min)}}^{X_{n(max)}} W(X_n)\, d\, X_n = 1.$$

The integral of the weighting density function

$$V(X_n) = \int_{X_{n(min)}}^{X_n} W(X_n)\, d\, X_n$$

is the weighting distribution function (Figure F-2).

Initially, the weighting density function is chosen uniformly with

$$W_{n(1)} = W_{n(2)} = , \dots , = W_{n(M_n-1)} .$$

The selection of subsequent starting points is weighted to reduce the probability of selecting points in the neighborhood of previously investigated points.

A random number R_n is generated for each independent variable, such that

$$0 \leq R_n \leq 1 .$$

The quantity

$$V(X_{n(m)}) = \int_{X_{n(min)}}^{X_{n(m)}} W(X_n)\, d\, X_n$$

is the value of the weighting distribution function at the m^{th} subdivision.

Then let q_n be the maximum value of m such that

$$R_n \geq V(X_{n(m)}) .$$

The initial value for the n^{th} independent variable is given by

$$X_{n,0} = X_{n(min)} + q_n \, DX_n + \frac{R_n - V(X_{n(g_n)})}{W(X_{n(g_n)}) \, DX_n} .$$

The weighted random starting point is then given by

$$(X_{1,0} , X_{2,0} , X_{3,0}, \ldots , X_{N,0}) .$$

3.1.2 Adjustment of Weights

Prior to the selection of the first starting point, the probabilities associated with all points are identical. However, after a starting point has been selected, the weight associated with the subinterval in which the point lies is reduced by a factor C, where $C > 1$. Hence, the probability of selecting subsequent points within the particular subinterval is reduced. The weight adjustment is given by

$$\frac{W_{n(m)}}{C} \rightarrow W_{n(m)} ; \; n = 1, 2, 3, \ldots , N ;$$

where m is such that

$$X_{n(m)} \leq X_{n,0} \leq X_{n(m+1)} .$$

3.1.3 Normalization of Weights

To effect the proper correspondence between the random number and the weighting distribution function in the generation of the probabilistic starting points, the weights of the density function must be normalized. To accomplish the normalization, the weight attached to each subinterval is divided by the product of the interval size (DX_n) and the summation of the weights over all M_n-1 subintervals.

$$\frac{W_{n(m)}}{\Delta X_n \sum_{m=1}^{M_n-1} W_{n(m)}} \rightarrow W_{n(m)}$$

for $m = 1, 2, 3, \ldots, M_n-1$ and

$n = 1, 2, 3, \ldots, N$.

Thus

$$DX_n \sum_{m=1}^{M_n-1} W_{n(m)} = 1$$

and

$$\int_{X_{n(min)}}^{X_{n(max)}} W(X_n)\, d X_n = 1 .$$

3.1.4 Region of Feasibility

The point from which the search for an extremum orginates must be in the feasible region. The feasible region (Figure F-1) may be defined as the N dimensional space which violates no boundaries of the independent variables nor any functional constraints imposed upon the variables. The boundaries of the independent variables are the minimum and maximum values imposed upon the variables. However, portions of the domain defined by the boundaries on the variables are often outside the feasible region. This condition can occur because of the functional constraints which restrict the independent variables.

The generation of a feasible starting point utilizes the previously described weighting procedure. A probabilitic starting point is generated and tested for feasibility. If the point is a feasible one, it becomes the base point from which the search for an extremum emanates. If the probabilistic starting point is not feasible, the weights associated with this point are adjusted to reduce the probability of selecting another point in this neighborhood. The weights are normalized and another probabilistic starting point is generated. The procedure is iterated until a feasible starting point (base point) is generated.

3.2 SEARCH DIRECTION

In considering the determination of the optimal direction of search for the local extremum (the extremum in a particular search direction), four situations must be investigated. The situations are the following:

Type 1 Search — The feasible base point is in no danger of violating boundaries or constraints (Figure F-3a).

(The increment ΔX_n is the basic increment and is given by

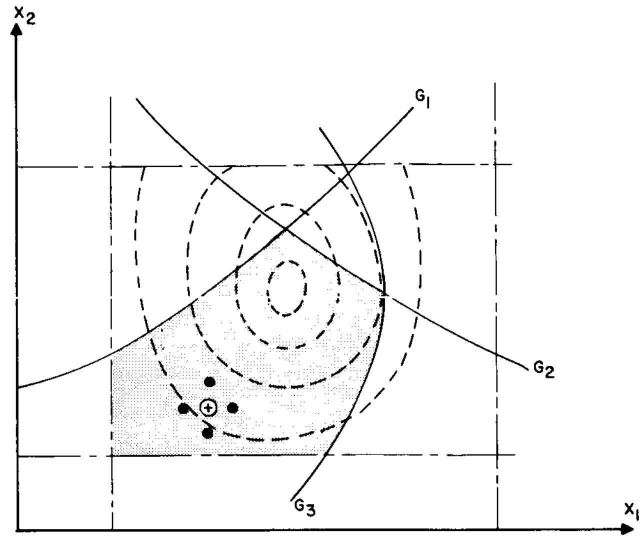
$$(X_{n(max)} - X_{n(min)}) \cdot A_n$$

where A_n is the accuracy (%) desired on the n^{th} independent variable.)

Type 2 Search — The feasible base point is within a ΔX_n of violating a constraint but from which a direction of search which minimizes the function can be computed by utilizing a procedure similar to the Type 1 Search (Figure F-3b).

Type 3 Search — The feasible base point is within a ΔX_n of violating a convex constraint and Type 1 and Type 2 Searches fail (Figure F-3c).

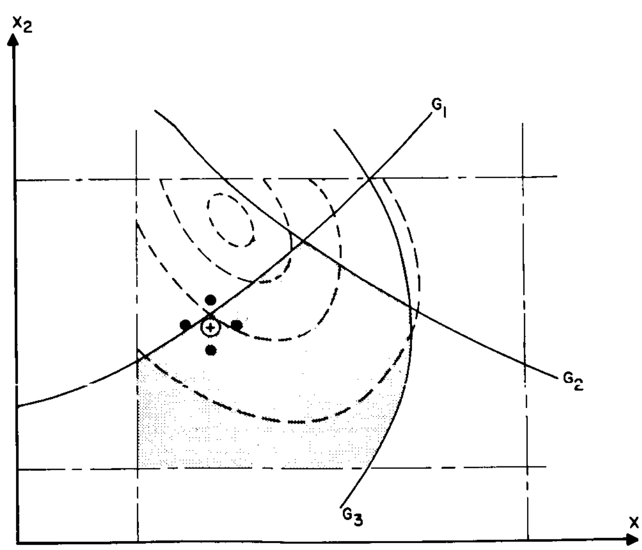
Type 4 Search — The feasible base point is within a ΔX_n of violating a concave constraint and Type 1, Type 2, and Type 3 Searches fail (Figure F-3d).



⊕ BASE POINT

● $\pm \Delta x_n$ FROM BASE POINT

Figure F-3a. Type 1 Search Situation



⊕ BASE POINT

● $\pm \Delta x_n$ FROM BASE POINT

Figure F-3b. Type 2 Search Situation

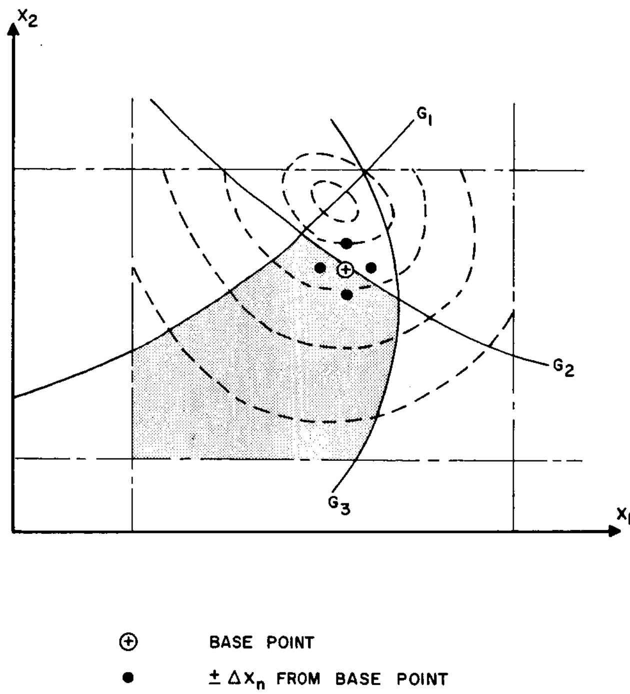


Figure F-3c. Type 3 Search Situation

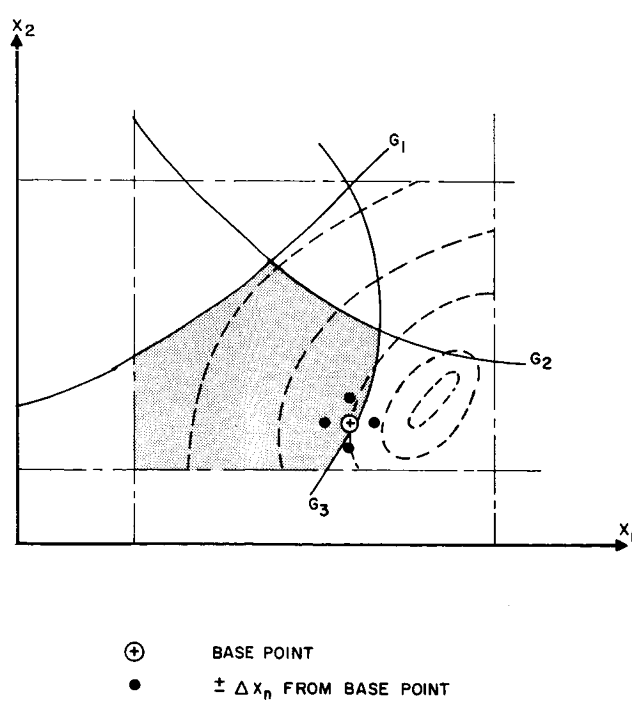


Figure F-3d. Type 4 Search Situation

The Search Vector is a vector emanating from the feasible base point in the direction in which the function approaches its extrenum in an optimal fashion, that is, in the direction along which the extrenum is approached most rapidly, but in which the constraints are satisfied for at least a unit step. The selection of the search vector reduces the search from a N-dimensional one to a one dimensional search along the search vector. Thus on the search vector, the dependent variable Y becomes an explicit function of only one independent variable λ, i. e.

$$Y = H(\lambda).$$

3.2.1 Type 1 Search

In the Type 1 Search the search vector is constructed by incrementing and decrementing each independent variable by a ΔX_n while maintaining the other independent variables constant at the base point. (See Figure F-4a). By evaluating the objective function at these points the search vector is determined as follows:

$$Y_0 = f(X_{1,0}, X_{2,0}, \dots, X_{N,0})$$

(function evaluated at base point)

$$Y_1^+ = f(X_{1,0} + \Delta X_1, X_{2,0}, \dots, X_{N,0})$$

$$Y_1^- = f(X_{1,0} - \Delta X_1, X_{2,0}, \dots, X_{N,0})$$

$$\vdots$$

$$Y_N^+ = f(X_{1,0}, X_{2,0}, \dots, X_{N,0} + \Delta X_N)$$

$$Y_N^- = f(X_{1,0}, X_{2,0}, \dots, X_{N,0} - \Delta X_N)$$

and calculate the following for each $n = 1, 2, \dots, N$

$$\Delta Y_n^+ = Y_0 - Y_n^+$$

$$\Delta Y_n^- = Y_0 - Y_n^-.$$

Note that $2n + 1$ function evaluations are required.

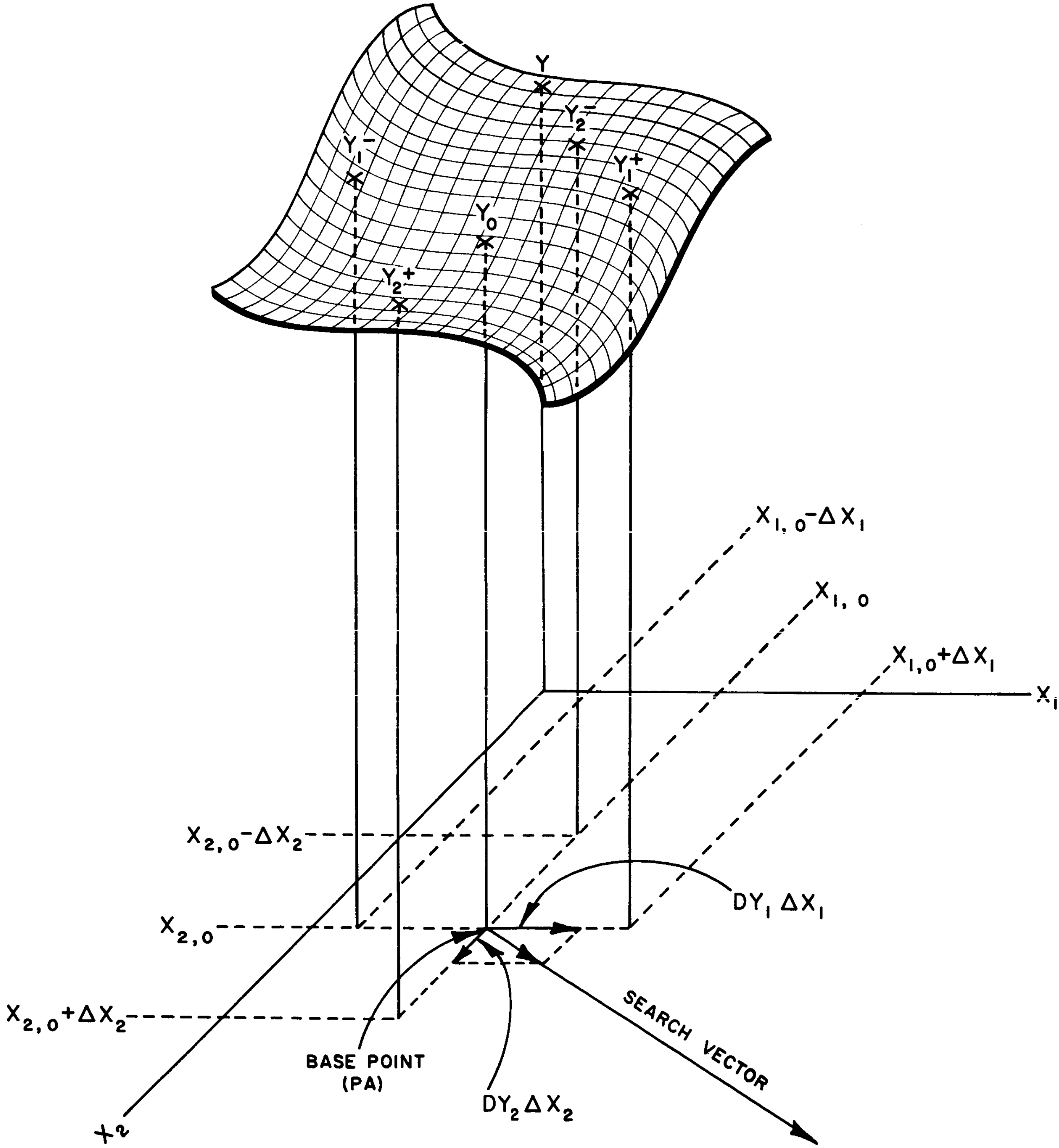


Figure F-4a. Type 1 Search Direction

Since the point which defines the maximum of the objective function $f(X_1, X_2, \ldots, X_N)$ is identical to that which defines the minimum of $-f(X_1, X_2, \ldots, X_N)$, the location of the direction of steepest descent is only considered.

If $\Delta Y_n^- \leq 0$ and $\Delta Y_n^+ \leq 0$ then $\Delta Y_n = 0$.

Otherwise let $\Delta Y_n^* = \text{maximum} \left[\Delta Y_n^- , \Delta Y_n^+ \right]$,

then if $\Delta Y_n^* = \Delta Y_n^-$, set $\Delta Y_n = - \left| \Delta Y_n^- \right|$

or if $\Delta Y_n^* = \Delta Y_n^+$, set $\Delta Y_n = \left| \Delta Y_n^+ \right|$.

Calculate the following for each $n = 1, 2, \ldots, N$

$$DY_n = \frac{\Delta Y_n}{\sqrt{\sum_{n=1}^{N} (\Delta Y_n)^2}}$$

where DY_n is a factor between -1 and 1 which is proportional to the rate at which the function approaches the extremum along the X_n axis at the base point.

The search vector is specified by a line from point P_A through point P_B where P_A is the base point and P_B is a point formed by incrementing each independent variable proportionately to the DY_n as follows:

$$P_A = (X_{1,0}, X_{2,0}, \ldots, X_{N,0})$$

and

$$P_B = \left[(X_{1,0} + DY_1 \cdot X_1), (X_{2,0} + DY_2 \cdot \Delta X_2), \ldots, (X_{N,0} + DY_N \cdot \Delta X_N) \right].$$

3.2.1.1 Test for Optimality

If $P_A = P_B$, the search vector is null, and consequently a relative extremum has been located. When this situation occurs, the search process is transferred to the weight normalization procedure, and another feasible base point is generated. However, if the search vector is not null, an arbitrary point on the search vector is given by

$$\left[(X_{1,0} + \lambda \cdot DY_1 \cdot \Delta X_1), (X_{2,0} + \lambda \cdot DY_2 \cdot \Delta X_2), \ldots, (X_{N,0} + \lambda \cdot DY_N \cdot \Delta X_N)\right].$$

Hence, the dependent variable Y becomes an explicit function of only one independent variable λ.

3.2.2 Type 2 Search

If the Type 1 Search fails, that is, a constraint is violated when a small step in the direction of steepest descent is taken, the Type 2 Search procedure is applied.

This procedure is similar to the Type 1 Search procedure. (See Figure F-4b). A search direction is computed by function evaluations at points formed by incrementing and decrementing each independent variable while maintaining the remaining variables constant at the feasible base point.

By evaluating the objective function and the constraining functions at these points the search direction is determined as follows:

Y_0 and $G_{0,j}$ are computed at the base point

$$Y_0 = f(X_{1,0}, X_{2,0}, \ldots, X_{N,0})$$

$$G_{0,j} = g_j(X_{1,0}, X_{2,0}, \ldots, X_{N,0}).$$

The function values at the adjusted points are

$$Y_1^+ = f(X_{1,0} + \Delta X_1, X_{2,0}, \ldots, X_{N,0})$$

$$G_{1,j}^+ = g_j(X_{1,0} + \Delta X_1, X_{2,0}, \ldots, X_{N,0})$$

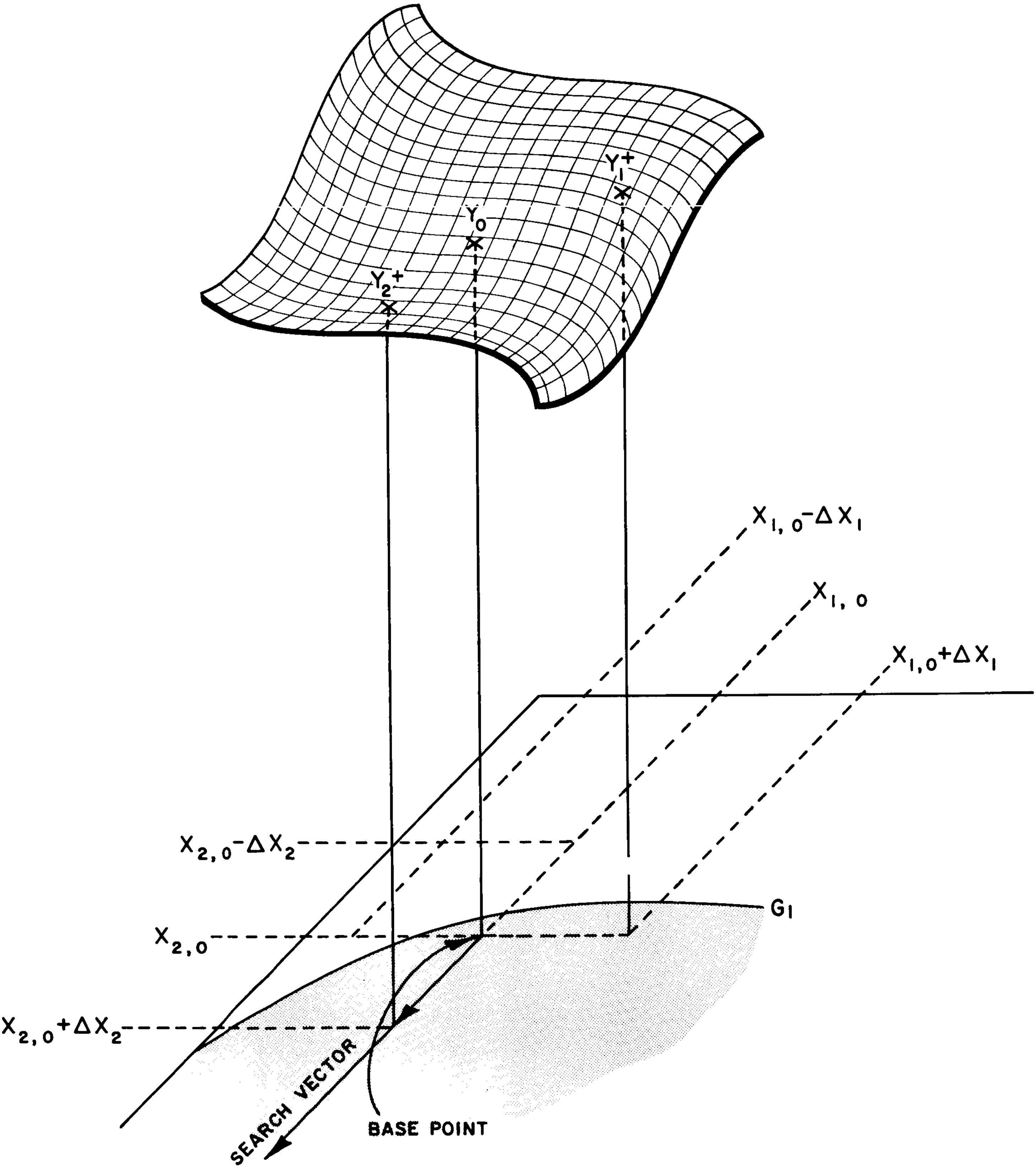


Figure F-4b. Type 2 Search Direction

$$Y_1^- = f\,(X_{1,0} - \Delta X_1,\ X_{2,0},\ \ldots,\ X_{N,0})$$

$$G_{1,j}^- = g_j\,(X_{1,0} - \Delta X_1,\ X_{2,0},\ \ldots,\ X_{N,0})$$

$$\vdots$$

$$Y_N^+ = f\,(X_{1,0},\ X_{2,0},\ \ldots,\ X_{N,0} + \Delta X_N)$$

$$G_{N,j}^+ = g_j\,(X_{1,0},\ X_{2,0},\ \ldots,\ X_{N,0} + \Delta X_N)$$

$$Y_N^- = f\,(X_{1,0},\ X_{2,0},\ \ldots,\ X_{N,0} - \Delta X_N)$$

$$G_{N,j}^- = g_j\,(X_{1,0},\ X_{2,0},\ \ldots,\ X_{N,0} - \Delta X_N)$$

Then calculate the following for each $n = 1, 2, \ldots, N$

$$\Delta Y_n^- = Y_0 - Y_n^-$$

if $G_{n,j}^- > 0$ for $j = 1, 2, \ldots, p$; or

$$\Delta Y_n^- = 0$$

if $G_{n,j}^- < 0$ at any $j = 1, 2, \ldots, p$ (constraint violation). G_j must be equal to or greater than 0 in order to remain in the feasible region.

Similarly,

$$\Delta Y_n^+ = Y_0 - Y_n^+$$

if $G_{n,j}^+ > 0$ for all j; or

$$\Delta Y_n^+ = 0$$

if $G_{n,j}^+ < 0$ for any j (constraint violation; Section 2.1).

Let $\Delta Y_n^* = \text{maximum} \left[\Delta Y_n^-, \; \Delta Y_n^+ \right]$.

If $\Delta Y_n^* = \Delta Y_n^-$, set $\Delta Y_n = - \left| \Delta Y_n^- \right|$.

If $\Delta Y_n^* = \Delta Y_n^+$, set $\Delta Y_n = \left| \Delta Y_n^+ \right|$.

Calculate for n = 1, 2, . . . , N

$$DY_n = \frac{\Delta Y_n}{\sqrt{\sum_{n=1}^{N} (\Delta Y_n)^2}}$$

where DY_n is a factor between -1 and 1. If $DY_n = 0$ for all n = 1, 2, . . ., N, the Type 2 Search procedure fails. If $DY_n \neq 0$ for all n = 1, 2, . . ., N, and no constraint is violated at a small step in the direction determined by the DY_n, a search vector is defined, and a search is made along this vector.

3.2.3 Type 3 Search

In the event that the previously described search directions are not valid (failures), the Type 3 Search* is used. (See Figure F-4c). In this case p' of the p constraints are violated when a unit step in the direction of steepest descent is taken.

As determined previously,

$$Y_0 = f\,(X_{1,0}, X_{2,0}, \ldots, X_{N,0})$$

$$G_{0,j} = g_j\,(X_{1,0}, X_{2,0}, \ldots, X_{N,0})$$

*H. Glass and L. Cooper, Sequential Search: A method for Solving Constrained Optimization Problems, ACM Journal Vol. 12, No. 1

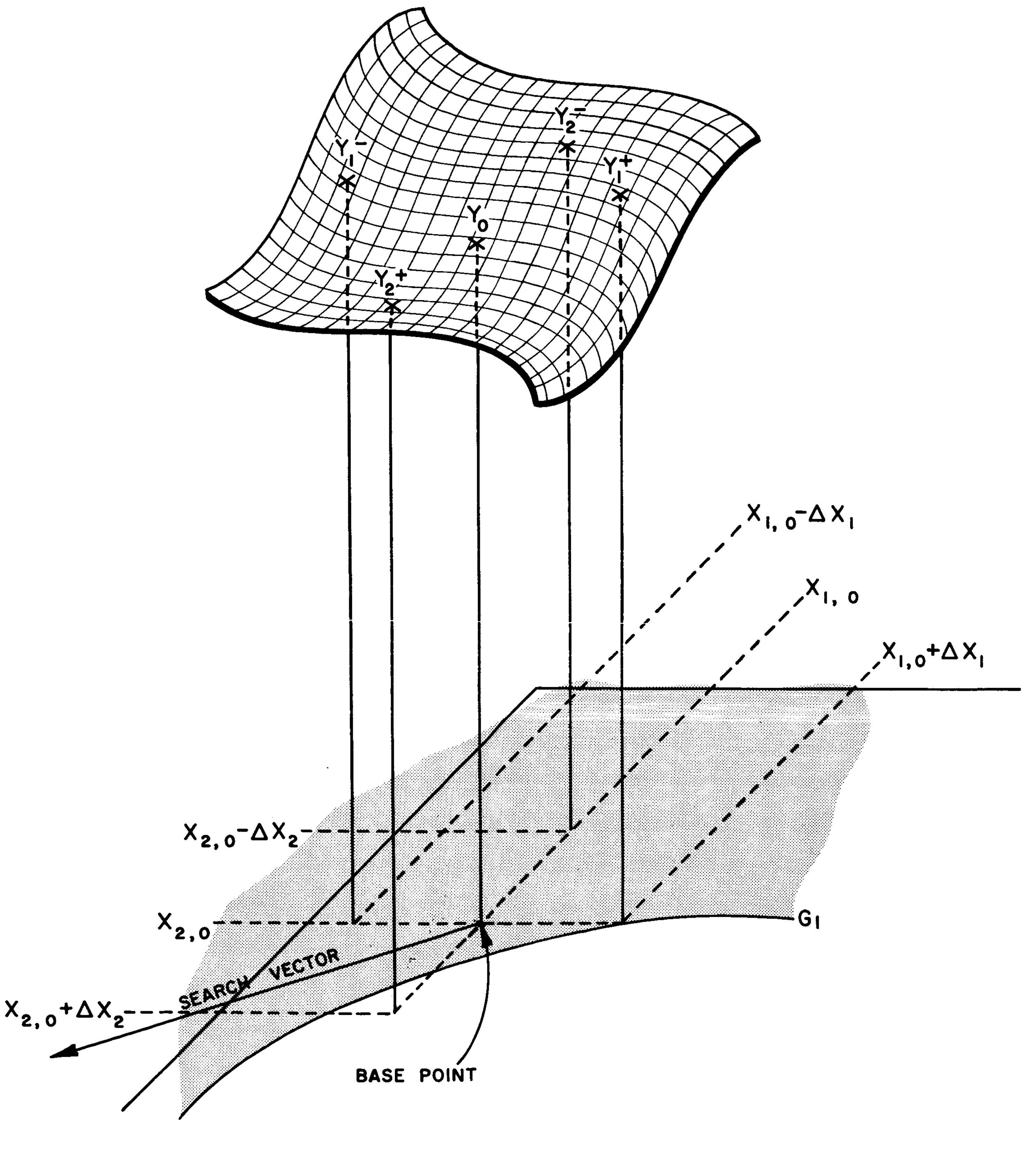


Figure F-4c. Type 3 Search Direction

$$Y_n^+ = f\,(X_{1,0},\ X_{2,0},\ \ldots,\ X_n + \Delta X_n,\ \ldots,\ X_{N,0})$$

$$G_{n,j}^+ = g_j\,(X_{1,0},\ X_{2,0},\ \ldots,\ X_n + \Delta X_n,\ \ldots,\ X_{N,0})$$

where $n = 1, 2, \ldots, N$

and $j = 1, 2, \ldots, p'$.

Then let

$$y_n = Y_n^+ - Y_0$$

and

$$h_{n,j} = G_{n,j}^+ - G_{0,j}$$

Suppose at the base point, some amount d_n is added to each variable. Considering a Taylor Series expansion of the objective function, where the second and higher degree terms are discarded, the value of the objective function at a new point is

$$f\,(X_{n,0}) + \sum_{n=1}^{N} \left.\frac{\partial f}{\partial X_n}\right|_{X_{n,0}} \cdot d_n \,.$$

However, since

$$\sum_{n=1}^{N} \frac{y_n}{\Delta X_n} \cong \sum_{n=1}^{N} \left(\frac{\partial f}{\partial X_n}\right),$$

the value becomes

$$f\,(X_{n,0}) + \sum_{n=1}^{N} \frac{y_n}{\Delta X_n} \cdot d_n \,.$$

Then a measure of the change in the value of the objective function is

$$\sum_{n=1}^{N} \frac{y_n d_n}{\Delta X_n} .$$

In like manner for the constraining functions, a measure of the changes in the G_j may be expressed as

$$\sum_{n=1}^{N} \frac{h_{n,j} \ d_n}{\Delta X_n}$$

for $j = 1, 2, \ldots, p'$ (p' constraints are violated if a unit step is taken in the direction determined by Type 1 Search). Since it is desired to minimize the objective function subject to p constraints, the search direction determined by d_n must be one in which the change in the objective function is negative and in which the changes in the constraining functions are greater than 0. Hence the following linear programming problem may be formulated.

Minimize

$$\sum_{n=1}^{N} y_n \ d_n$$

subject to

$$\sum_{n=1}^{N} h_{n,j} \ d_n \geq 0$$

and

$$-1 \leq d_n \leq 1$$

where

$$j = 1, 2, \ldots, p' .$$

However, the linear programming problem demands that $d_n \geq 0$. Thus let $t_n = d_n + 1$ for $n = 1, 2, \ldots, N$ and reformulate the linear problem.

Hence,

minimize

$$\sum_{n=1}^{N} y_n t_n$$

subject to

$$\sum_{n=1}^{N} h_{n,j} t_n$$

and

$$0 \leq t_n \leq 2$$

for $j = 1, 2, \ldots, p'$.

The solution vector t_n may be found by utilizing the Simplex Method. * The vector $t_n - 1$ is the direction vector in which the search is performed.

3.2.4 Type 4 Search

In some cases where the base point is very near a concave constraint, all three previously described procedures for determining a search direction fail. In this situation the technique described in the following discussion is applied. (See Figure F-4d.)

The direction vector (d_n)† computed in the Type 3 Search procedure is retained.

The p constraints are investigated to determine which constraints are violated at a small move in the direction given by the d_n.

*Saul I. Gass, "Linear Programming Methods and Applications"; McGraw-Hill; 1964

† $d_n = t_n^{-1}$

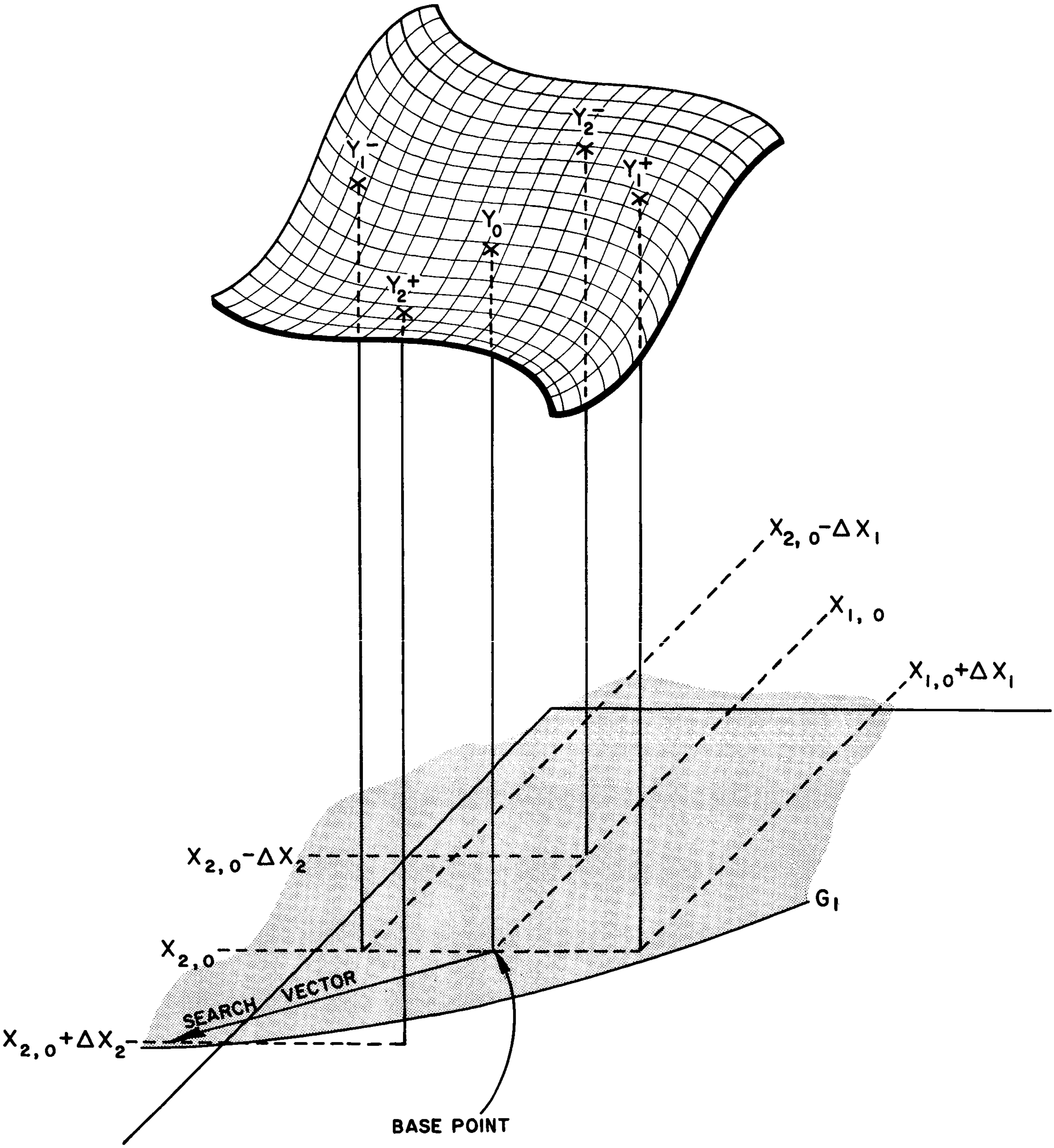


Figure F-4d. Type 4 Search Direction

$$G'_j = g_j \left[(X_{1,0} + \lambda \cdot \Delta X_1 \cdot d_1), (X_{2,0} + \lambda \cdot \Delta X_2 \cdot d_2), \ldots, (X_{N,0} + \lambda \cdot \Delta X_N \cdot d_N) \right]$$

$$0 < \lambda < 1$$

$$j = 1, 2, \ldots, p.$$

The violating constraints are then compared in order to determine the one which is most negative at the small step;

$$G^* = \text{minimum} \left[G'_j \right]$$

$$j = 1, 2, \ldots, p'.$$

Then determine the direction which maximizes G* using the Type 1 Search procedure. Let DG_n be the direction vector determined by the Type 1 Search. A new base point is determined by moving a unit step ($\lambda = 1$) in the direction along the vector determined by the DG_n . The new base point is

$$\left[(X_{1,0} + \lambda \cdot \Delta X_1 \cdot DG_1), (X_{2,0} + \lambda \cdot \Delta X_2 \cdot DG_2), \ldots, (X_{N,0} + \lambda \cdot \Delta X_N \cdot DG_N) \right.$$

where

$$\lambda = 1.$$

The search is then resumed from the new base point in the direction determined by the retained d_n vector.

3.3 DETERMINATION OF THE LOCAL EXTREMUM ALONG THE SEARCH VECTOR

The determination of a local extremum along the search vector includes the following:

- Expanding Search
- Contracting Search.

3.3.1 Expanding Search

The Expanding Search (Figure F-5) determines the approximate location of the objective function extremum along the search vector by evaluating the function at successive points along the search vector. These points are specified in the following manner:

$$Y_k = f\Big[(X_{1,0} + \lambda_k \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + \lambda_k \cdot \Delta X_2 \cdot DY_2), \ldots, (X_{N,0} + \lambda_k \cdot \Delta X_N \cdot DY_N)\Big]$$

where

$$\lambda_k = \gamma^{k-1} - 1 ;$$

$$k = 1, 2, \ldots, u ;$$

$$\gamma > 1 ;$$

and u is the value of k when the expanding search is terminated.

The expanding search is terminated, and the search process is transferred to the contracting search along the same search vector when

$$\Delta Y' = Y_K - Y_{K-1} \geq 0$$

or when the constraint evaluated at the k^{th} step

$$G_{K,j} < 0 \qquad (j = 1, 2, \ldots, p) .$$

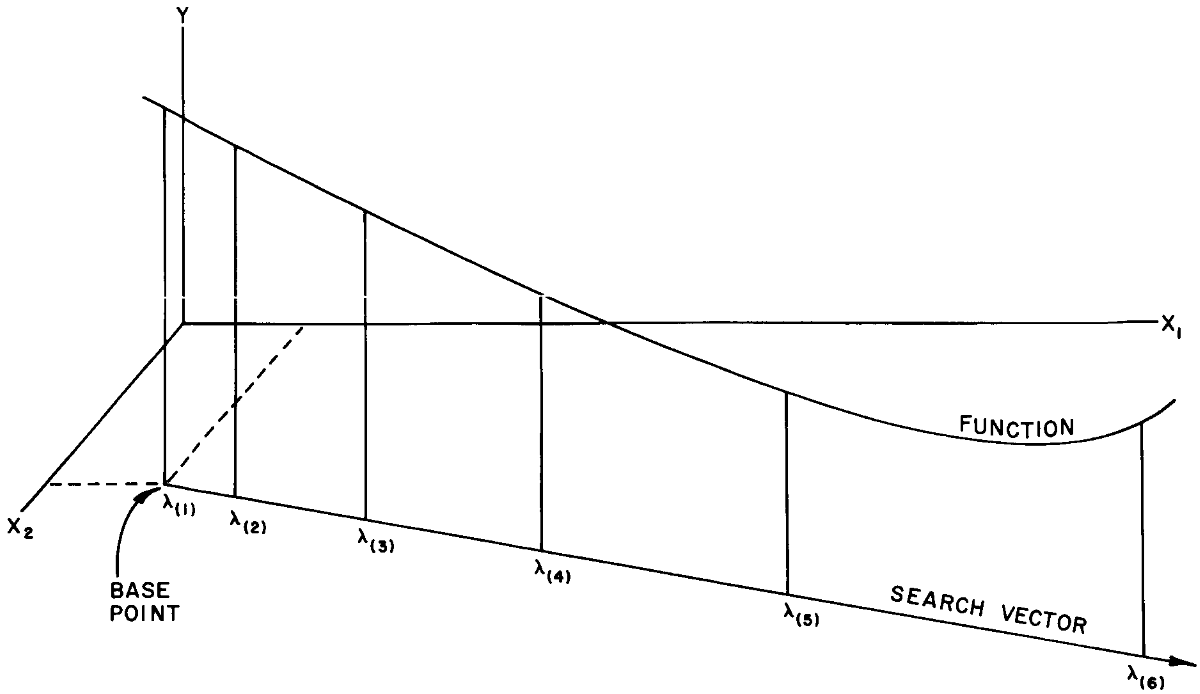


Figure F-5. Expanding Search

The values of λ $(\lambda^-, \lambda^0, \lambda^+)$ which confine the location of the extremum are given by

$$\lambda^- = \lambda_{(u-2)}$$

$$\lambda^0 = \lambda_{(u-1)}$$

$$\lambda^+ = \lambda_{(u)} \ .$$

If neither of the above conditions for termination is satisfied before a boundary is exceeded, the value of $\lambda_{boundary}$ is calculated at the boundary.

A new feasible base point is then given by

$$X_{n,0} = \left[(X_{1,0} + \lambda_{boundary} \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + \lambda_{boundary} \cdot \Delta X_2 \cdot DY_2), \ldots, (X_{N,0} + \lambda_{boundary} \cdot \Delta X_N \cdot DY_N) \right].$$

3.3.2 Contracting Search

The Contracting Search (Figure F-6) further confines the location of the local extremum along the search vector within the range defined by the expanding search and in the feasible region.

The technique utilized is a simple halving procedure. Two additional points are determined by halving the span between the points determined by λ^-, λ^0, and λ^+.

$$T\lambda^- = \frac{\lambda^- + \lambda^0}{2}$$

$$T\lambda^+ = \frac{\lambda^+ + \lambda^0}{2}$$

The objective function and constraining functions are evaluated at the additional points along with the λ^0 point.

$$Y(T\lambda^-) = f\left[(X_{1,0} + T\lambda^- \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + T\lambda^- \cdot \Delta X_2 \cdot DY_2), \ldots, (X_{N,0} + T\lambda^- \cdot \Delta X_N \cdot DY_N) \right]$$

$$Y(\lambda^0) = f\left[(X_{1,0} + \lambda^0 \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + \lambda^0 \cdot \Delta X_2 \cdot DY_2), \ldots, (X_{N,0} + \lambda^0 \cdot \Delta X_N \cdot DY_N) \right]$$

$$Y(T\lambda^+) = f\left[(X_{1,0} + T\lambda^+ \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + T\lambda^+ \cdot \Delta X_2 \cdot DY_2), \ldots, (X_{N,0} + T\lambda^+ \cdot \Delta X_N \cdot DY_N) \right].$$

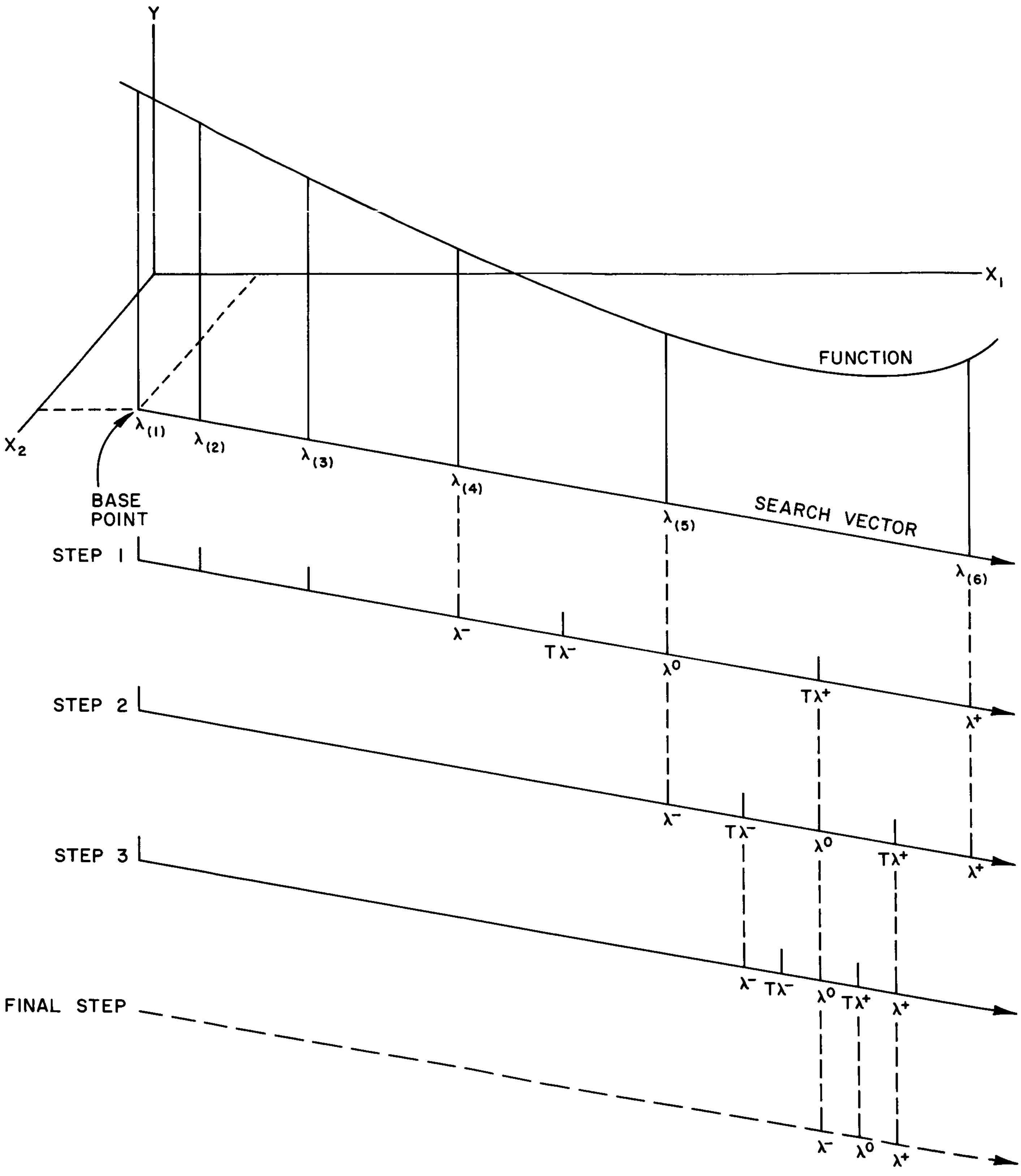


Figure F-6. Contracting Search

Similarly

$$G_j(T\lambda^-) = g_j\Big[(X_{1,0} + T\lambda^- \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + T\lambda^- \cdot \Delta X_2 \cdot DY_2),$$
$$\ldots, (X_{N,0} + T\lambda^- \cdot \Delta X_N \cdot DY_N)\Big]$$

$$G_j(\lambda^0) = g_j\Big[(X_{1,0} + \lambda^0 \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + \lambda^0 \cdot \Delta X_2 \cdot DY_2),$$
$$\ldots, (X_{N,0} + \lambda^0 \cdot \Delta X_N \cdot DY_N)\Big]$$

$$G_j(T\lambda^+) = g_j\Big[(X_{1,0} + T\lambda^+ \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + T\lambda^+ \cdot \Delta X_2 \cdot DY_2),$$
$$\ldots, (X_{N,0} + T\lambda^+ \cdot \Delta X_N \cdot DY_N)\Big]$$

for $j = 1, 2, \ldots, p$.

The minimum of $Y(T\lambda^-)$, $Y(\lambda^0)$, $Y(T\lambda^+)$ redefines a new λ^0 if a constraint is not violated at any of these three points. If a constraint is violated at any of the three points, the minimum of $Y(T\lambda^-)$, $Y(\lambda^0)$, $Y(T\lambda^+)$ which does not violate a constraint defines the new λ^0. The point adjacent to the new λ^0 in the negative direction is redefined λ^- and likewise the positive adjacent point is λ^+.

These new λ's define a reduced range on the search vector. This halving procedure is iterated until

$$\lambda^+ - \lambda^0 \leq \left|\frac{1}{DY_n}\right|$$

where $n = 1, 2, \ldots, N$.

This insures the confinement on each of the independent variables to be less than ΔX_n. Since the increment on the independent variables corresponding to the increment $\Delta\lambda = (\lambda^+ - \lambda^0)$ is given by

$$BX_n = (\lambda^+ - \lambda^0) \cdot DY_n \cdot \Delta X_n$$

where

$$\lambda^+ - \lambda^0 \leq \left| \frac{1}{DY_n} \right|$$

then

$$BX_n \leq \Delta X_n .$$

Since the local extremum is now located within the required accuracy (ΔX_n), the contracting search is completed and a new feasible base point is defined. The new feasible base point is

$$X_{n,0} = \Big[(X_{1,0} + \lambda^0 \cdot \Delta X_1 \cdot DY_1), (X_{2,0} + \lambda^0 \cdot \Delta X_2 \cdot DY_2), \ldots, (X_{N,0} + \lambda^0 \cdot \Delta X_N \cdot DY_N)\Big].$$

Having located the local extremum on the search vector, the weights associated with the point are adjusted (Section 3.1.2) and a new search direction is determined (Section 3.2). The entire procedure is iterated until the required number of relative extrema have been located. From these relative extrema the global extremum is selected.

4 SUMMARY AND CONCLUSION

Isolating that combination of system parameter values which extremize some system characteristic (cost, reliability, performance, etc.) is recognized as system optimization*. Mathematically, this operation is equivalent to maximizing or minimizing a function of the system parameters which describe the system characteristic being considered. Unfortunately, the complexity of modern systems often prohibits the utilization of analytical methods. In these cases the use of a high-speed digital computer is required.

The computer technique developed at LSI to determine the optimal system configuration within the constraints imposed on the system is the Generalized Random Extremum Analysis Technique. GREAT, as the routine is called, can perhaps best be described as an iterative scheme based upon a combination of the direct search and gradient techniques to which has been added a learning capability.

The search for the extrema of the function to be optimized originates at a point chosen probabilistically in the N dimensional feasible space determined by the N system parameters. The direction of steepest descent at the starting point is computed by utilizing the function values at positive and negative increments along each of the independent variable axes.

The sub-optimal point along the direction of steepest descent is determined by sequential function evaluations in the direction of steepest descent. If the condition for optimality is not satisfied at the sub-optimum, a new direction for search is determined as above. The procedure is iterated until the condition for optimality is met and a relative extremum is found. The iteration of the technique to generate a number of relative extrema coupled with the learning scheme provide a statistical confidence of selecting the true global extremum from the relative extrema.

4.1 MODULARITY

It is quite well known in the field of optimization that no single technique developed thus far is able to solve efficently a broad class of optimization problems. As new and better techniques for solving the problem are developed, an efficient means for the transition from the technique stage to the working program stage is desirable. With this

* The non-linear optimization problem may be thought of as the extremizing of a non-linear objective function which can be constrained by inequality functions which can be linear, non-linear, or a combination of the two types.

thought in mind, the GREAT program was organized in a modular manner. Hence, the facility for the insertion of new techniques and for the interchange of existing techniques is apparent. See Figure F-7.

4.2 DISCRETE VALUED VARIABLES

Since the independent variables in many physical systems are not continuous, that is they can assume only discrete values on a range, the objective functions describing these systems, though continuous analytically, exist physically only at discrete points in the N dimensional space. If the number of variables and the number of discrete values which each variable can assume are small, the optimal solution can be obtained quite easily by an exhaustive evaluation of the function at all points defined by the combination of discrete variables. However, if these numbers become large, the solution becomes very time consuming. For example, a ten-variable problem with each variable possessing ten discrete values would require 10^{10} function evaluations and 10^{10} comparisons to determine the optimal solution.

A method has been proposed, the merits of which are not fully known, which should accomplish this end of discrete optimization. Very briefly this method of discrete optimization can be described in the following manner. A constrained optimal solution of the N independent variable problem is found using some continuous variable technique. The values of the independent variables obtained in the continuous treatment are compared to the array of allowed discrete values. The discrete value of the variable which matches its corresponding continuous variable from the optimal set most closely without a constraint violation is selected as the variable to be inserted as a constant in the objective and constraining functions for further computation. Thus the N dimensional problem becomes an N-1 dimensional one. This procedure is repeated until the optimization problem becomes unidimensional at which time the problem is solved.

The procedure possesses the capacity for utilization in cases in which both discrete and continuous variables are present in the same problem.

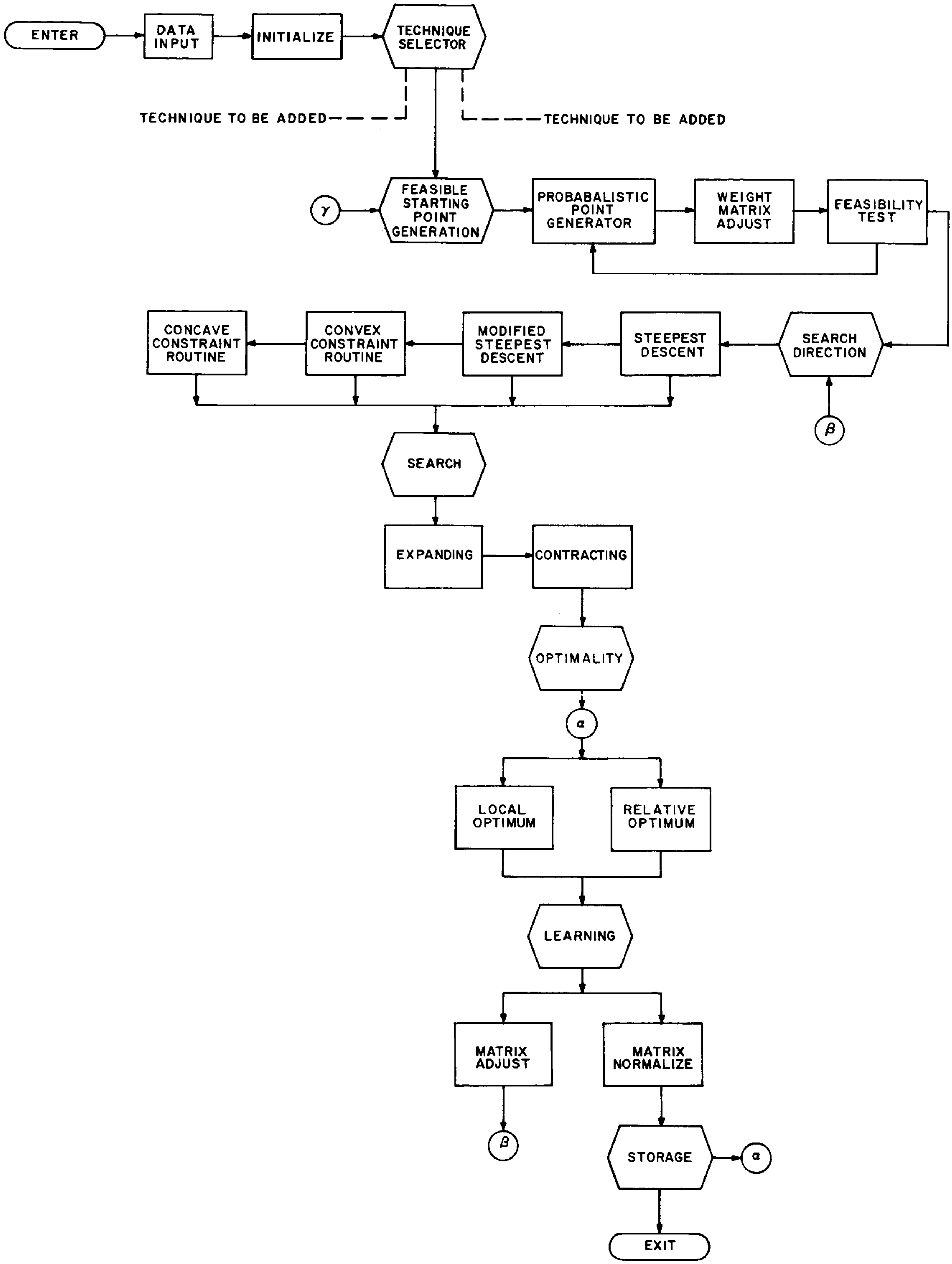


Figure F-7. Modulator Construction

5 OPERATION

Considerable effort was directed toward the development of a user-oriented program. In the area of input, Data Transmittal Forms were developed on which the data required for analysis are listed and from which the data are directly prepared for computer utilization. Section 5.1 describes these forms in detail.

The GREAT output is also in a form which is easily interpreted. The output consists mainly of a listing of the input information and the solution to the optimization problem. Section 5.2 presents an example problem with its output.

The subroutine (FUNCY), which evaluates the objective function and which tests the constraining functions, must be written and compiled each time a different problem is optimized. The writing of the subroutine requires some knowledge of FORTRAN II.

5.1 DATA TRANSMITTAL INSTRUCTIONS (See Figure F-8 and Table F-I)

Data Transmittal Forms have been developed on which the required data for obtaining an analysis are listed and from which the data are directly prepared for computer utilization. The utilization of these forms requires only a familiarity with the GREAT methodology. These Data Transmittal Forms (DTF) are identified by a GREAT DTF number located in the title block of each form.

Explicit instructions are presented here for utilization of these forms. It might be well to note that these forms consist of blocks which are 80 columns wide. These columns are representative of an eighty-column punch card. Each row on the DTF represents a card. It also might be well to define four terminologies that are used in the instructions with which the reader may not be familiar. These are: (1) floating point number, (2) left justification, (3) right justification, (4) alpha-numeric characters. A floating point number may be defined as one with a decimal point at the beginning, at the end, or between any two digits of the number. Left justification is the positioning of the left-most alpha-numeric character in the left-most position of the field. Similarly, right justification is the positioning of the right-most alpha-numeric character in the right-most position of the field. Alpha-numeric characters consist of all the alphabetic characters (A through Z), all the numeric characters (0 through 9), and the special characters ([.], [,], [+], [-], [/], [*], [$], [=], [(], [)]).

LSI	**LEAR SIEGLER, INC.** INSTRUMENT DIVISION	GREAT DTF I	**FUNCTION EXTREMUM ANALYSIS**	SHEET ____ OF ____

PROBLEM DESCRIPTION CARDS

NAME OF ORIGINATOR | PROBLEM NAME | DATE

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

1

EXTREMUM | GAMMA | X | C | E | S | WEIGHT | I | R

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

2

NS | FUNCTION ANALYZED

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

3
3
3
3
3
3
3
3
3
3
3

Figure F-8. Sample Data Transmittal Form I

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
2 thru 15	NAME OF ORIGINATOR	Alpha-numeric	Left Justified	Name of person who requested the analysis
21 thru 60	PROBLEM NAME	Alpha-numeric	No Restrictions	Name of problem under consideration
71 and 72 74 and 75 77 and 78	DATE	Integer	Right Justified	Date: Month Day Year
80	None	1	Pre-Assigned	Control Number

CARD 2

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
6 thru 12	EXTREMUM	MINIMUM or MAXIMUM	As Specified	Type of extremum desired
18 thru 22	GAMMA	Flating Point Number Greater Than One	Left Justified	Controls rate of expansion in the Expanding Search; Typical values are 1.5 and 2.0
28 and 29	X	Integer	Right Justified	Number of independent variables
35 and 36	C	Integer	Right Justified	Number of constraining functions
42 and 43	E	Integer	Right Justified	Number of Relative Extremums desired
49 and 50	S	Integer	Right Justified	Number of Secondary Searches; Typical values 10 to 90
56 thru 60	WEIGHT	Floating Point Number	Left Justified	Weighting Factor: Typical values 2 to 5
66 and 67	I	0 or 1	Right Justified	Indicator of intermediate output (0 indicates no intermediate output) (1 indicates intermediate output)
73 and 74	R	0 or 1	Right Justified	Indicator of read-in starting point (0 indicates random points) (1 indicates read-in points)
80	None	2	Pre-assigned	Control Number

CARD 3

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
1 thru 3	NS	Integer	Right Justified	Number of cards required to describe the problem
80	None	3	Pre-assigned	Control Number

CARD 4 thru 13

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
1 thru 72	Function Analyzed	Alpha-numeric	Not Restricted	Description of the problem in conventional symbols or in Fortran Code

Table F-I. Data Transmittal Form I

5.1.1 Data Transmittal Form I

Data Transmittal Form I describes the optimization problem to GREAT. The problem description includes the following information:

First Card

. NAME OF ORIGINATOR:	The NAME OF ORIGINATOR is the name of the person who requested the analysis.
. PROBLEM NAME:	The PROBLEM NAME is the name associated with the problem under consideration.
. DATE:	The DATE represents the time at which the analysis was performed.

Second Card

. EXTREMUM:	The EXTREMUM represents the type of extremum sought, either a MINIMUM or MAXIMUM.
. GAMMA:	The GAMMA represents the expansion factor in the Expanding Search.
. X :	The X represents the number of independent variables.
. C :	The C represents the number of constraints to the optimization problem.
. E :	The E represents the number of Relative Extremums desired.
. S :	The S represents the number of secondary searches allowed (number of search directions).
. WEIGHT:	The WEIGHT represents the weighting factor. (Ref. 3.1.2.)
. I :	The I indicates whether intermediate output is desired (0, No; 1, Yes).

. R : The R indicates whether the starting point is to be read in (0, No; 1, Yes).

Third Card: Function Analyzed Block

. NS : The NS represents the number of cards required to describe the function and constraints.

. FUNCTION ANALYZED : The FUNCTION ANALYZED represents the function to be analyzed in conventional symbols or in FORTRAN CODE.

5.1.2 Data Transmittal Forms II, IIA, and IIB (See Figure F-9 and Table F-II)

Data Transmittal Forms II, IIA, and IIB describe the independent variables to GREAT. The description of the independent variables includes the following:

Independent Variable Cards

. VN : The VN represents an Independent Variable Number which corresponds to a particular independent variable in the analysis.

. NAME: The NAME represents the name assigned to the particular independent variable.

. Range MINIMUM: The MINIMUM represents the lower bound of the independent variable value.

. Range MAXIMUM: The MAXIMUM represents the upper bound of the independent variable value.

. INITIAL VALUE: The INITIAL VALUE represents the starting value of the independent variable if called for by R in Card 2 of DTFI.

LSI	LEAR SIEGLER, INC. INSTRUMENT DIVISION	GREAT DTF II	FUNCTION EXTREMUM ANALYSIS	SHEET ____OF____

INDEPENDENT VARIABLE CARDS

VN (1–2)	3–5	NAME (6–11)	12–16	RANGE MINIMUM (17–26)	27–31	RANGE MAXIMUM (32–41)	42–46	INITIAL VALUE (47–56)	57–61	PERCENT ERROR (62–67)	68–72	DIV (73–76)	77–79	80
1														4
2														4
3														4
4														4
5														4
6														4
7														4
8														4
9														4
10														4
11														4
12														4
13														4
14														4
15														4
16														4
17														4
18														4
19														4
20														4

Figure F-9. Sample Data Transmittal Form II, IIA, IIB (1 of 3)

LSi	LEAR SIEGLER, INC. INSTRUMENT DIVISION	GREAT DTF IIA	FUNCTION EXTREMUM ANALYSIS	SHEET ____ OF ____

INDEPENDENT VARIABLE CARDS

VN	NAME	RANGE MINIMUM	RANGE MAXIMUM	INITIAL VALUE	PERCENT ERROR	DIV	80
21							4
22							4
23							4
24							4
25							4
26							4
27							4
28							4
29							4
30							4
31							4
32							4
33							4
34							4
35							4
36							4
37							4
38							4
39							4
40							4

Figure F-9. Sample Data Transmittal Form II, IIA, IIB (2 of 3)

LSI	LEAR SIEGLER, INC. INSTRUMENT DIVISION	GREAT DTF IIB	FUNCTION EXTREMUM ANALYSIS	SHEET ____ OF ____

INDEPENDENT VARIABLE CARDS

VN (1–2)	NAME (6–11)	RANGE MINIMUM (17–26)	RANGE MAXIMUM (32–41)	INITIAL VALUE (47–56)	PERCENT ERROR (62–67)	DIV (73–76)	80
41							4
42							4
43							4
44							4
45							4
46							4
47							4
48							4
49							4
50							4
51							4
52							4
53							4
54							4
55							4
56							4
57							4
58							4
59							4
60							4

Figure F-9. Sample Data Transmittal Form II, IIA, IIB (3 of 3)

Table F-II. Data Transmittal Form II, IIA, IIB

COLUMNS	HEADING	ALLOWED CHARACTERS	ALLOWED FORMAT	DESCRIPTION
1 and 2	VN	Integer	Pre-Assigned	Number which identifies the independent variable
6 thru 11	NAME	Alpha-numeric	Left Justified	Name of independent variable (volts, amps, gyro, etc.)
17 thru 26	MINIMUM	Floating Point Number	No Restrictions	Lower limit on the range of the values of the independent variable
32 thru 41	MAXIMUM	Floating Point Number	No Restrictions	Uper limit on the range of the values of the independent variable
47 thru 56	INITIAL VALUE	Floating Point Number	No Restrictions	Starting value of the independent variable (if called for by DTFI)
62 thru 67	PERCENT ERROR	Floating Point Number	No Restrictions	Allowable error in percent of the specified range
73 thru 76	DIV	Integer	Right Justified	Number of intervals on the independent variable (typical value = 10)
80	None	4	Pre-Assigned	Control Number

- PERCENT ERROR: The PERCENT ERROR represents the percent of the range error allowed.

- DIV: The DIV represents the number of intervals into which the independent variable is divided. This information is used both by the feasible base point selection routine and by learning routines.

5.2 EXAMPLE

The example problem selected for presentation in this appendix is a four independent variable problem constrained by three functional inequalities. The particular problem was formulated by J. B. Rosen and S. Suzuki and appeared in the February, 1965 issue of "Communications of the ACM". It was chosen because it is constrained by concave functions and hence is a difficult problem.

The problem is:

Maximize*

$$f\,(x_n) = -X_1^2 - X_2^2 - 2X_3^2 - X_4^2 + 5X_1 + 5X_2 + 21X_3 - 7X_4$$

subject to the following constraints

$$G_1 = -X_1^2 - X_2^2 - X_3^2 - X_4^2 - X_1 + X_2 - X_3 + X_4 + 8 \geq 0$$

$$G_2 = -X_1^2 - 2X_2^2 - X_3^2 - 2X_4^2 + X_1 + X_4 + 10 \geq 0$$

$$G_3 = -2X_1^2 - X_2^2 - X_3^2 - 2X_1 + X_2 + X_4 + 5 \geq 0$$

*The problem in the "Communications of the ACM" reads "minimize" the function. However, it is felt that this is a result of an oversight on the part of the authors or a misinterpretation by the reader.

The optimal solution vector is

$$X_1 = 0$$

$$X_2 = 1$$

$$X_3 = 2$$

$$X_4 = -1$$

yielding an objective function value of 44.

The Data Transmittal Forms on which the required information is listed are included together with a listing of the user written subroutine FUNCY and the output. (Figures F-10, F-11, and F-12, respectively.)

LEAR SIEGLER, INC. INSTRUMENT DIVISION	GREAT DTF I	FUNCTION EXTREMUM ANALYSIS	SHEET 1 OF 2

PROBLEM DESCRIPTION CARDS

NAME OF ORIGINATOR	PROBLEM NAME	DATE	Card (col. 80)
NASA NASW-938	ROSEN - SUZUKI EXAMPLE PROBLEM		1

EXTREMUM	GAMMA	X	C	E	S	WEIGHT	I	R	Card (col. 80)
MAXIMUM	1.4	4	3	2	25	2.0	1	0	2

NS	FUNCTION ANALYZED	Card (col. 80)
6		3
	Y=-X(1)*X(1)-X(2)*X(2)-2.*X(3)*X(3)-X(4)*X(4)+5.*X(1)+5.*X(2)+21.*	3
	1X(3)-7.*X(4)	3
	1 G(1)=-X(1)*X(1)-X(2)*X(2)-X(3)*X(3)-X(4)*X(4)-X(1)+X(2)-X(3)+X(4)+	3
	18.	3
	2 G(2)=-X(1)*X(1)-2.*X(2)*X(2)-X(3)*X(3)-2.*X(4)*X(4)+X(1)+X(4)+10.	3
	3 G(3)=-2 *X(1)*X(1)-X(2)*X(2)-X(3)*X(3)-2.*X(1)+X(2)+X(4)+5.	3
		3
		3
		3
		3

Figure F-10. Example Problem Data Transmittal Forms (1 of 2)

LEAR SIEGLER, INC. INSTRUMENT DIVISION	GREAT DTF II	FUNCTION EXTREMUM ANALYSIS	SHEET 2 OF 2

INDEPENDENT VARIABLE CARDS

VN	NAME	RANGE MINIMUM	RANGE MAXIMUM	INITIAL VALUE	PERCENT ERROR	DIV	80
1	X(1)	-3.0	3.0		1.	10	4
2	X(2)	-3.0	3.0		1.	10	4
3	X(3)	-3.0	3.0		1.	10	4
4	X(4)	-3.0	3.0		1.	10	4
5							4
6							4
7							4
8							4
9							4
10							4
11							4
12							4
13							4
14							4
15							4
16							4
17							4
18							4
19							4
20							4

Figure F-10. Example Problem Data Transmittal Forms (2 of 2)

```
      SUBROUTINE FUNCY(IEXT,X,Y,IC,C,IFRT,NCON)
      DIMENSION X(50),IC( 5),C( 5)
      IF(IFRT-40)10,10,20
   10 Y=-X(1)*X(1)-X(2)*X(2)-2.*X(3)*X(3)-X(4)*X(4)+5.*X(1)+5.*X(2)+21.*
     1X(3)-7.*X(4)
      IF(IEXT-1)20,23,20
   23 Y=-Y
   20 IP=IFRT-40
      DO 11 I=1,NCON
      GO TO (1,2,3),I
    1 C(1)=-X(1)*X(1)-X(2)*X(2)-X(3)*X(3)-X(4)*X(4)-X(1)+X(2)-X(3)+X(4)+
     18.
      GO TO 6
    2 C(2)=-X(1)*X(1)-2.*X(2)*X(2)-X(3)*X(3)-2.*X(4)*X(4)+X(1)+X(4)+10.
      GO TO 6
    3 C(3)=-2.*X(1)*X(1)-X(2)*X(2)-X(3)*X(3)-2.*X(1)+X(2)+X(4)+5.
    6 IF(C(I))7,8,8
    8 IC(I)=0
      GO TO 11
    7 IC(I)=1
   11 CONTINUE
      IF(IFRT-40)21,21,22
   21 DO 12 I=1,NCON
      IF(IC(I))13,12,13
   12 CONTINUE
      IFRT=2
      GO TO 15
   13 IFRT=3
   15 RETURN
   22 Y=-C(IP)
      GO TO 15
      END
```

Figure F-11. Subroutine FUNCY for Example Problem

```
GREAT-M5              *** FUNCTION EXTREMUM ANALYSIS ***        PAGE  1 OF  3
NASA NASW-938          ROSEN - SUZUKI EXAMPLE PROBLEM           DATE  9/ 9/65
                                  MAXIMUM
*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
                          ** RELATIVE EXTREMUMS **
         NUMBER      DEPENDENT              INDEPENDENT      SYMBOL
                      VARIABLE                VARIABLES

           1      Y= .4391E 02
                                        X( 1)=-.3609E-01     VAR. 1
                                        X( 2)= .1012E 01     VAR. 2
                                        X( 3)= .1980E 01     VAR. 3
                                        X( 4)=-.9918E 00     VAR. 4
           2      Y= .4387E 02
                                        X( 1)=-.9970E-01     VAR. 1
                                        X( 2)= .1023E 01     VAR. 2
                                        X( 3)= .2060E 01     VAR. 3
                                        X( 4)=-.9082E 00     VAR. 4
```

```
GREAT-M5              *** FUNCTION EXTREMUM ANALYSIS ***        PAGE  2 OF  3
NASA NASW-938          ROSEN - SUZUKI EXAMPLE PROBLEM           DATE  9/ 9/65
                                  MAXIMUM
*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
                          ** FUNCTION ANALYZED **

   Y=-X(1)*X(1)-X(2)*X(2)-2.*X(3)*X(3)-X(4)*X(4)+5.*X(1)+5.*X(2)+21.*
  1X(3)-7.*X(4)
 1 G(1)=-X(1)*X(1)-X(2)*X(2)-X(3)*X(3)-X(4)*X(4)-X(1)+X(2)-X(3)+X(4)+
  18.
 2 G(2)=-X(1)*X(1)-2.*X(2)*X(2)-X(3)*X(3)-2.*X(4)*X(4)+X(1)+X(4)+10.
 3 G(3)=-2.*X(1)*X(1)-X(2)*X(2)-X(3)*X(3)-2.*X(1)+X(2)+X(4)+5.

*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
                                   NOTES.
```

```
GREAT-M5              *** FUNCTION EXTREMUM ANALYSIS ***        PAGE  3 OF  3
NASA NASW-938          ROSEN - SUZUKI EXAMPLE PROBLEM           DATE  9/ 9/65
                                  MAXIMUM
*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*
                ** INDEPENDENT VARIABLE DESCRIPTIONS **
      VARIABLE      SYMBOL     ---------RANGE---------         PERCENT
        NAME                    MINIMUM       MAXIMUM            ERROR

      VAR. 1        X( 1)     -.3000E 01     .3000E 01           1.00
      VAR. 2        X( 2)     -.3000E 01     .3000E 01           1.00
      VAR. 3        X( 3)     -.3000E 01     .3000E 01           1.00
      VAR. 4        X( 4)     -.3000E 01     .3000E 01           1.00
```

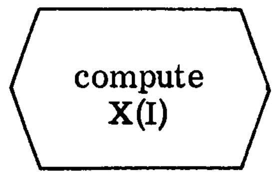
Figure F-12. Output for Example Problem

6 PROGRAM

This section contains a complete set of logic diagrams (Figure F-13) along with listings of the GREAT program (Figure F-14). The logic diagrams serve as an interface between the text and the program listings. Hence, the reader ought to be able to understand basically the mechanism of GREAT.

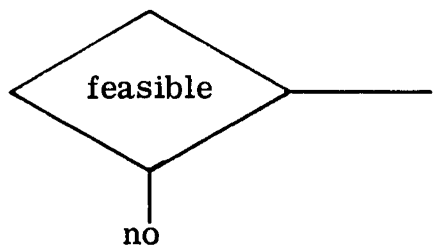
These diagrams present the logic of the program by statements enclosed by symbols. The five symbols used are:

1) the process symbol



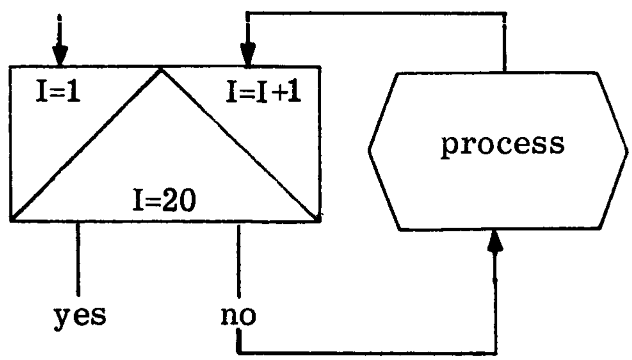
;

2) the decision symbol



;

3) the iteration symbol



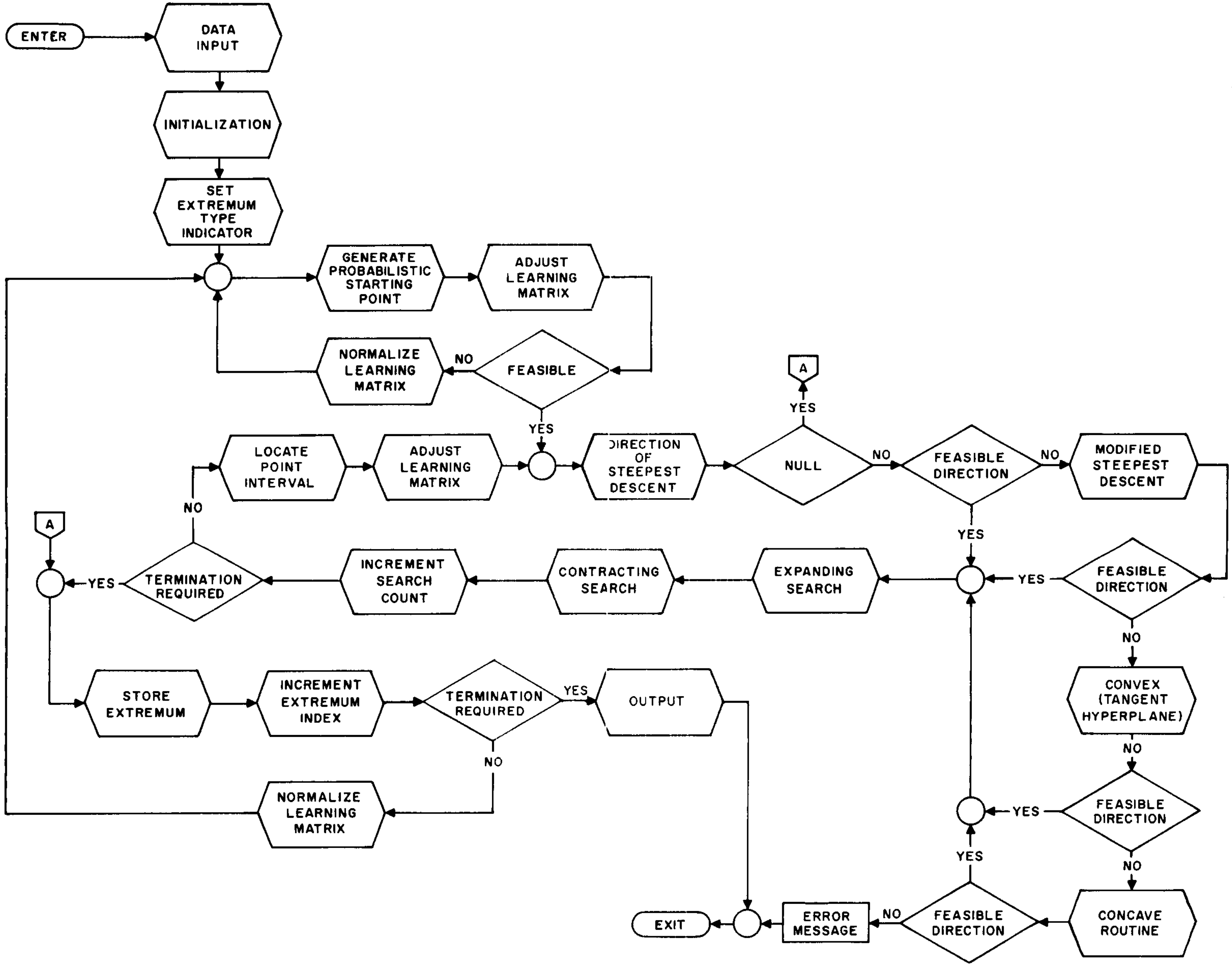
4) the direction symbol

; and

5) the terminal symbol

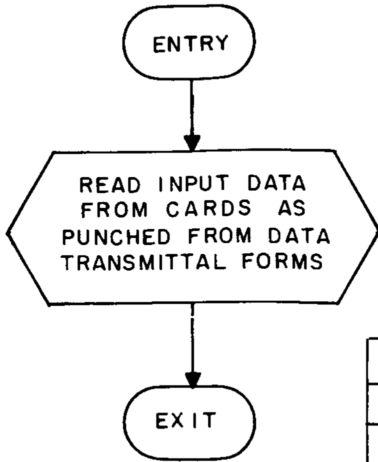
.

The GREAT program has been written in the FORTRAN II language. The particular version listed in this appendix is an IBM 7090 version using the University of Michigan Executive System. The library subroutines used in the program are described in the University of Michigan Executive System Manual. As an additional aid comment cards have been inserted to describe the program variables and various program operations.



Mainline (GREAT)

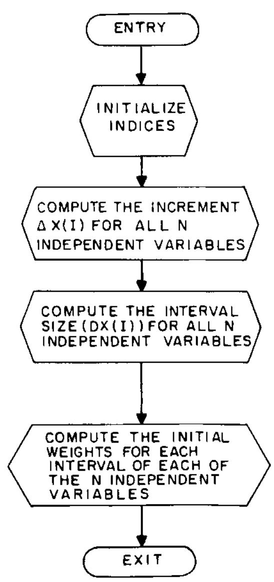
Figure F-13. Optimization Logic Diagrams (1 of 21)



ARGUMENTS	
INPUT	OUTPUT
	NMAX
	GAMMA
	LPMAX
	LSMAX
	WFAC
	NCON
	INDIC
	NW
	XIPT

Data Input (DATA) Subroutine

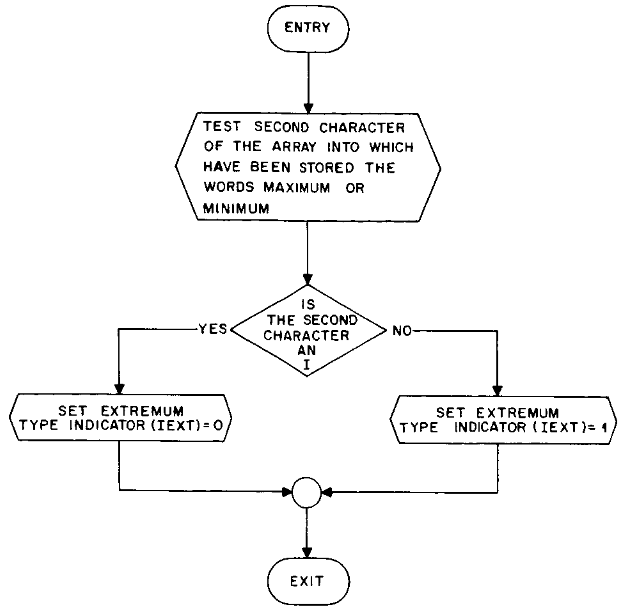
Figure F-13. Optimization Logic Diagrams (2 of 21)



ARGUMENTS	
INPUT	OUTPUT
NMAX	IFRT
XMAX	LS
XMIN	LP
ACC	MARY
NW	LAMB
	DX
	DDX
	WGT

Initialization (START) Subroutine

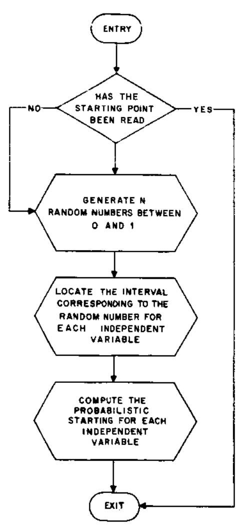
Figure F-13. Optimization Logic Diagrams (3 of 21)



ARGUMENTS	
INPUT	OUTPUT
ITYPE	IEXT

Extremum Type (EXTREM) Subroutine

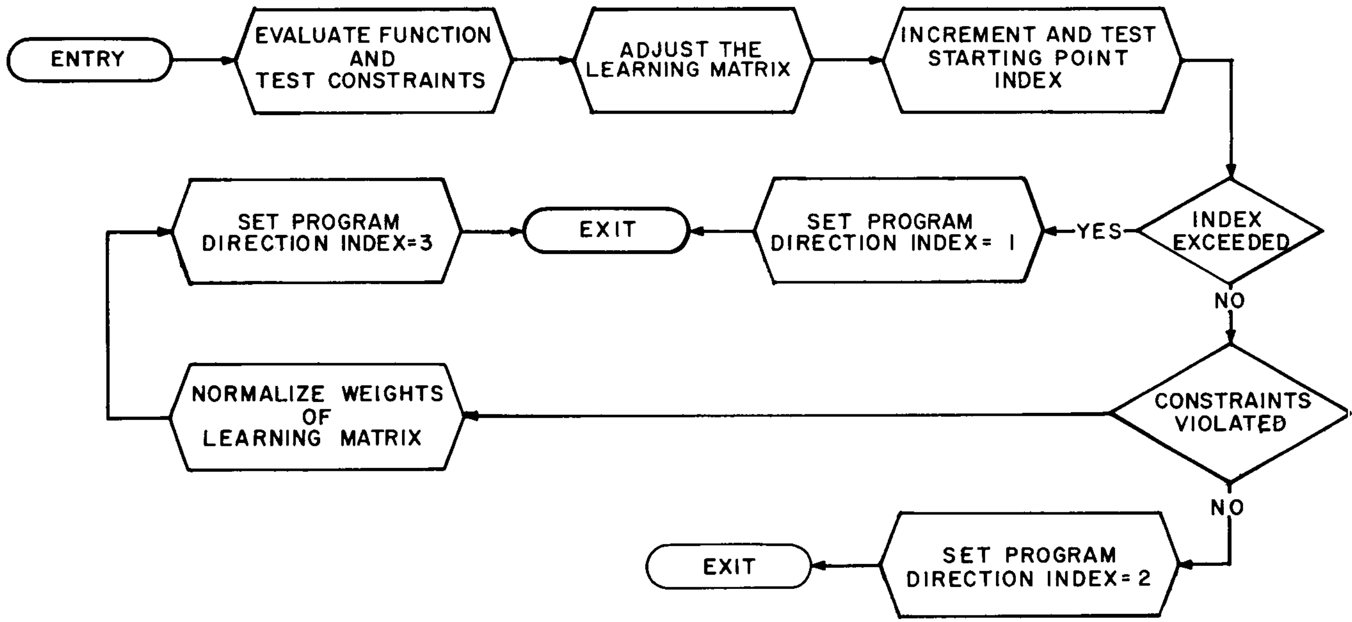
Figure F-13. Optimization Logic Diagrams (4 of 21)



ARGUMENTS	
INPUT	OUTPUT
INDIC	XIPT
NMAX	JI
NW	
WGT	
DDX	
XMIN	

Probabilistic Point Generating (NPOINT) Subroutine

Figure F-13. Optimization Logic Diagrams (5 of 21)

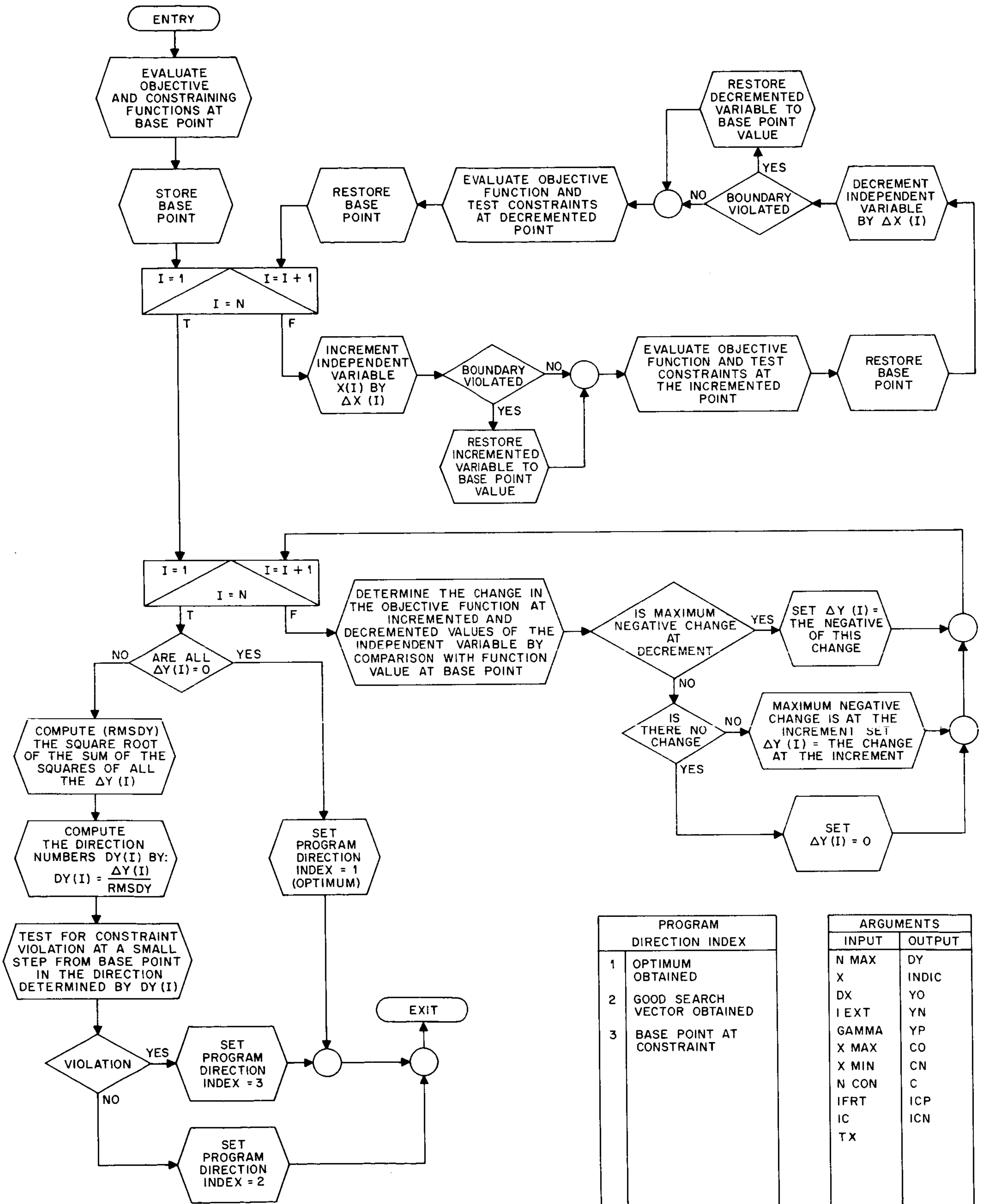


ARGUMENTS	
INPUT	OUTPUT
MARY LAMB IEXT XIPT IC Z IFRT NCON NMAX WGT WFAC JI NW DDX	INDIC

PROGRAM DIRECTION INDEX	
1	UNABLE TO GENERATE FEASIBLE STARTING
2	CURRENT POINT IS FEASIBLE
3	CURRENT IS NOT FEASIBLE

Feasibility Test (FEASBL) Subroutine

Figure F-13. Optimization Logic Diagrams (6 of 21)



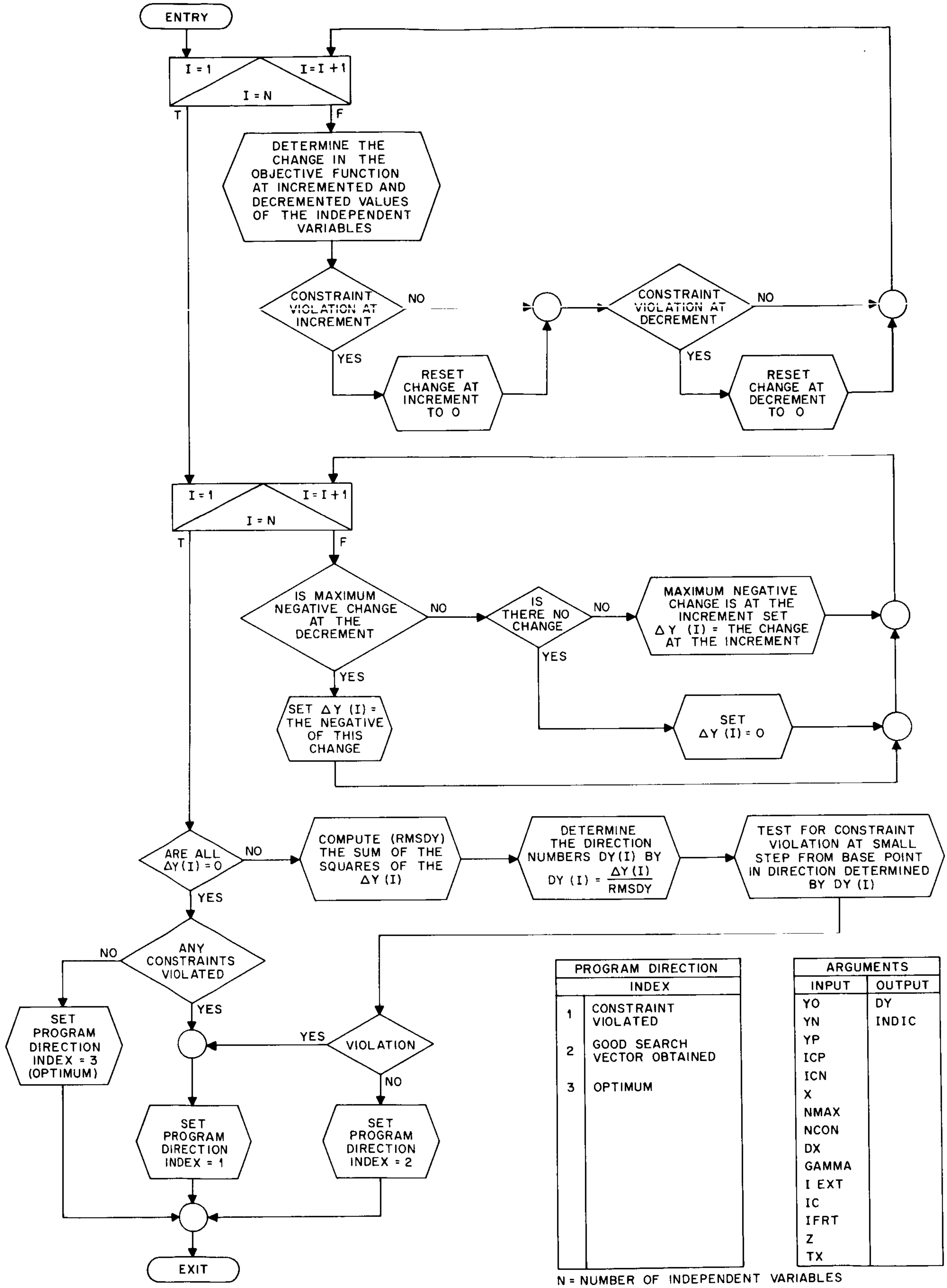
PROGRAM DIRECTION INDEX	
1	OPTIMUM OBTAINED
2	GOOD SEARCH VECTOR OBTAINED
3	BASE POINT AT CONSTRAINT

ARGUMENTS	
INPUT	OUTPUT
N MAX	DY
X	INDIC
DX	YO
I EXT	YN
GAMMA	YP
X MAX	CO
X MIN	CN
N CON	C
IFRT	ICP
IC	ICN
TX	

N = NUMBER OF INDEPENDENT VARIABLES

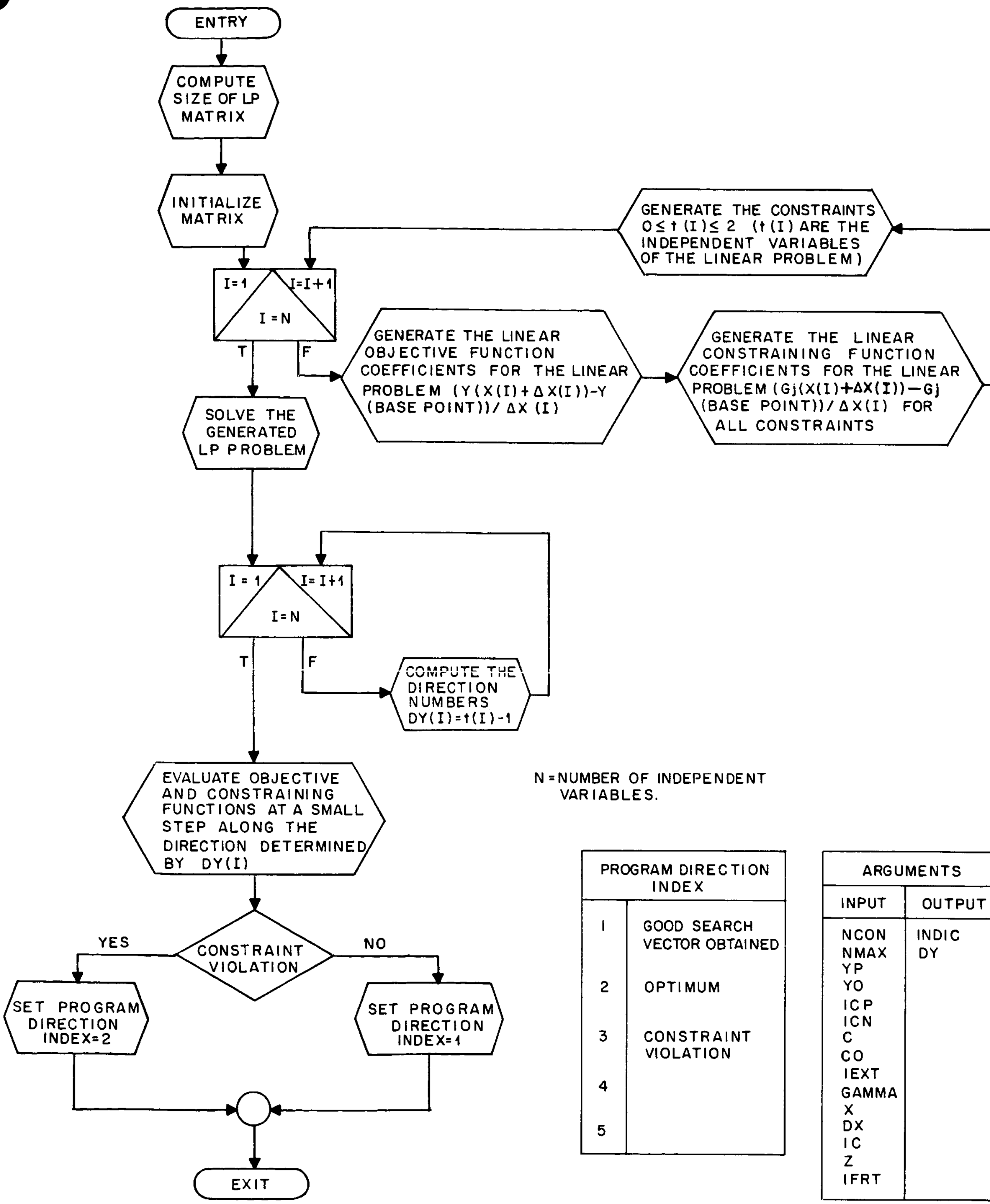
Type 1 Search (STEEP) Subroutine

Figure F-13. Optimization Logic Diagrams (7 of 21)



Type 2 Search (STEEP1) Subroutine

Figure F-13. Optimization Logic Diagrams (8 of 21)



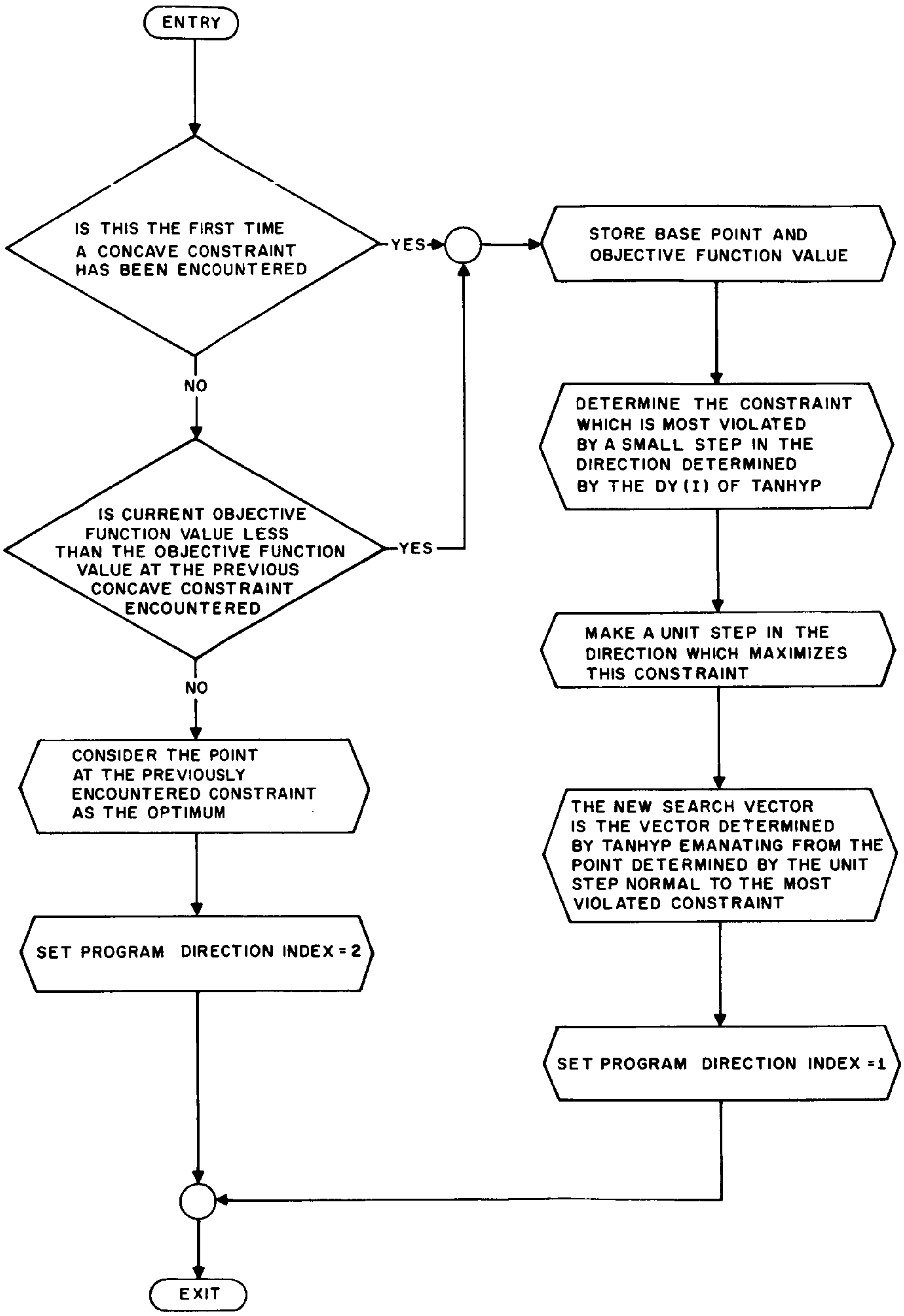
N = NUMBER OF INDEPENDENT VARIABLES.

PROGRAM DIRECTION INDEX	
1	GOOD SEARCH VECTOR OBTAINED
2	OPTIMUM
3	CONSTRAINT VIOLATION
4	
5	

ARGUMENTS	
INPUT	OUTPUT
NCON NMAX YP YO ICP ICN C CO IEXT GAMMA X DX IC Z IFRT	INDIC DY

Type 3 Search (TANHYP) Subroutine

Figure F-13. Optimization Logic Diagrams (9 of 21)

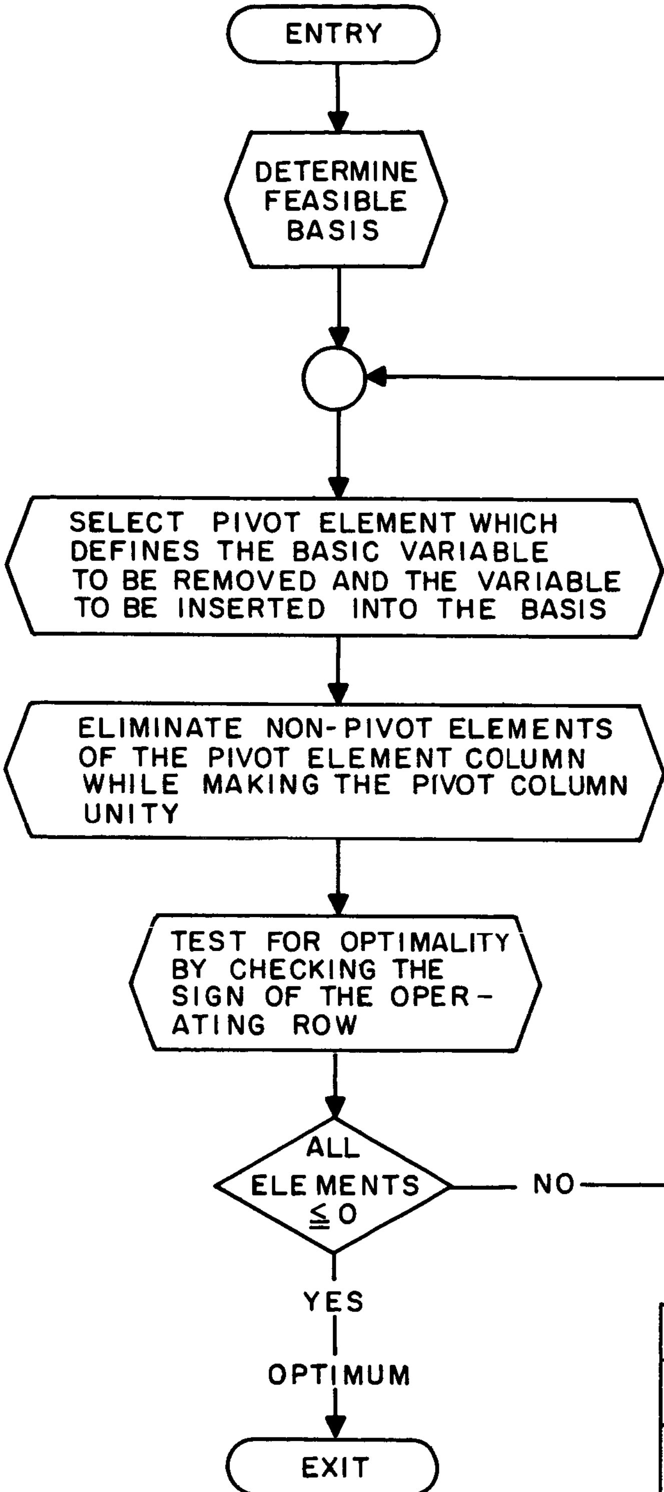


ARGUMENTS	
INPUT	OUTPUT
KONCAV	KONCAV
YO	DY
NMAX	INDIC
X	
NCON	
Z	
IEXT	
GAMMA	
DX	
XMAX	
XMIN	
YN	
YP	
CO	
CN	
C	
ICP	
ICN	
IFRT	
IC	
TX	

PROGRAM DIRECTION INDEX	
1	GOOD SEARCH VECTOR OBTAINED
2	OPTIMUM
3	
4	
5	

Type 4 Search (TANP1) Subroutine

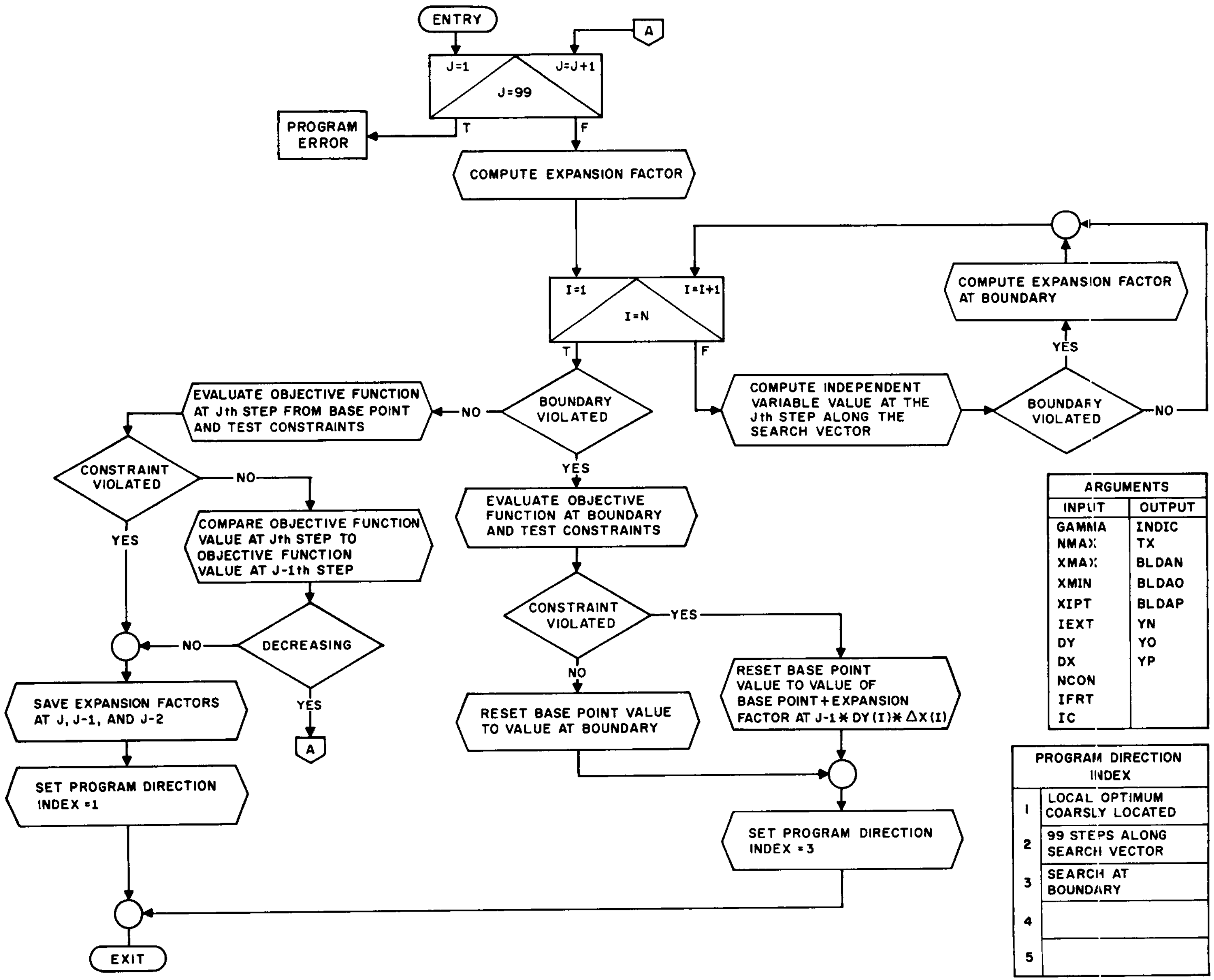
Figure F-13. Optimization Logic Diagrams (10 of 21)



ARGUMENTS	
INPUT	OUTPUT
II JJ A IEXT NMAX	LSUB DY

Linear Programming (LPSUB) Subroutine

Figure F-13. Optimization Logic Diagrams (11 of 21)

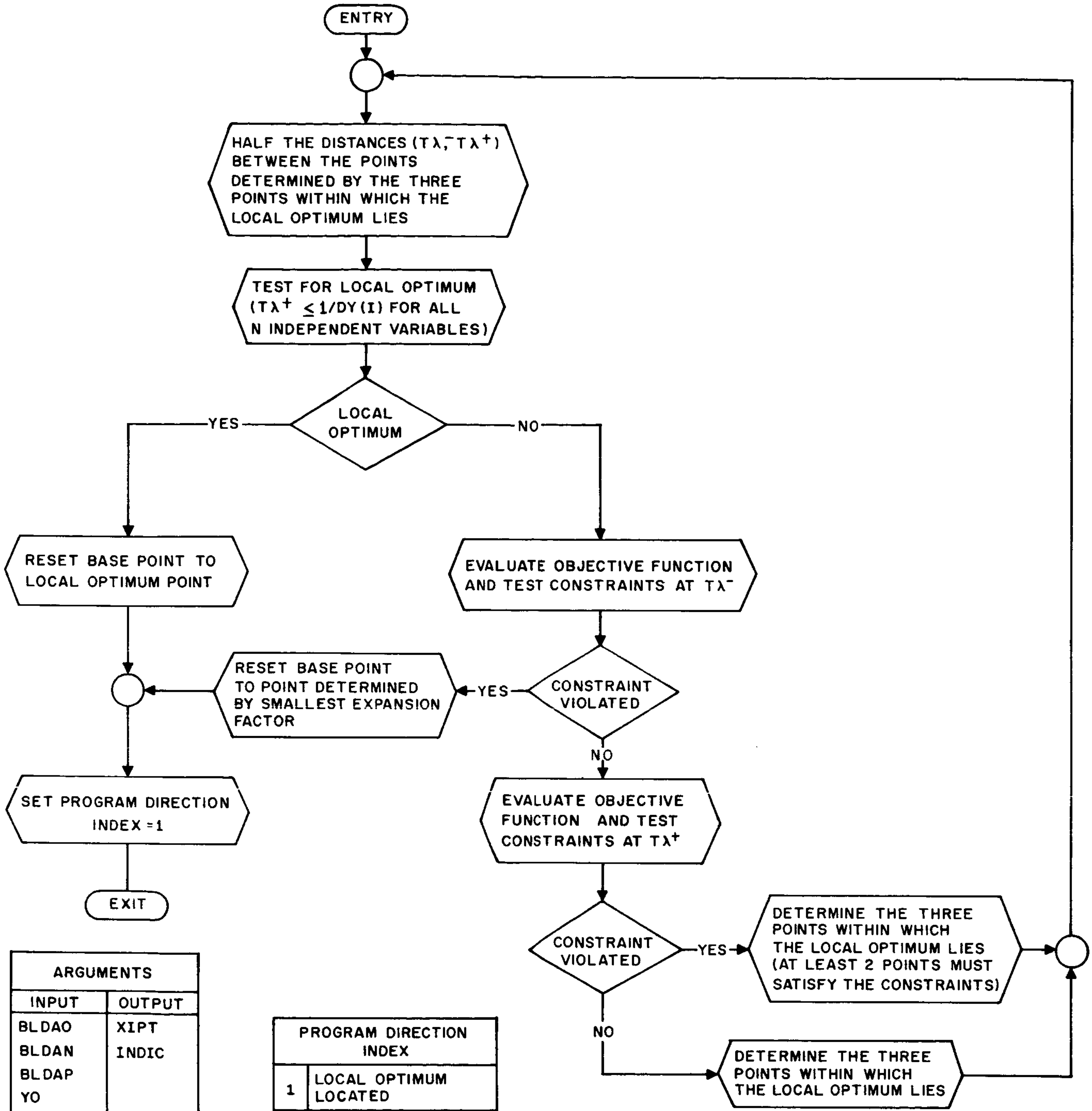


ARGUMENTS	
INPUT	OUTPUT
GAMMA	INDIC
NMAX	TX
XMAX	BLDAN
XMIN	BLDAO
XIPT	BLDAP
IEXT	YN
DY	YO
DX	YP
NCON	
IFRT	
IC	

PROGRAM DIRECTION INDEX	
1	LOCAL OPTIMUM COARSLY LOCATED
2	99 STEPS ALONG SEARCH VECTOR
3	SEARCH AT BOUNDARY
4	
5	

Expanding Search (EXPAND) Subroutine

Figure F-13. Optimization Logic Diagrams (12 of 21)

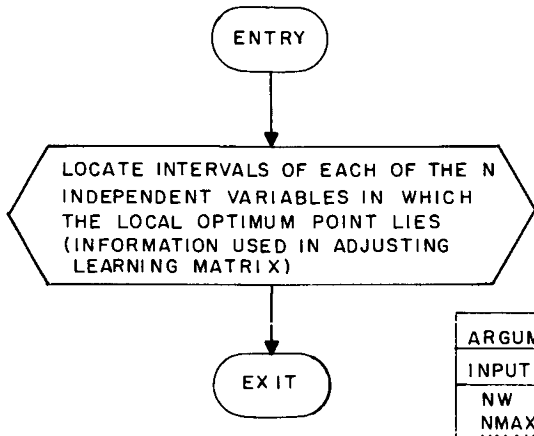


ARGUMENTS	
INPUT	OUTPUT
BLDAO	XIPT
BLDAN	INDIC
BLDAP	
YO	
YN	
YP	
NMAX	
IEXT	
TX	
DY	
DX	
NCON	
IC	
Z	
IFRT	

PROGRAM DIRECTION INDEX	
1	LOCAL OPTIMUM LOCATED
2	
3	
4	
5	

Contracting Search (NARROW) Subroutine

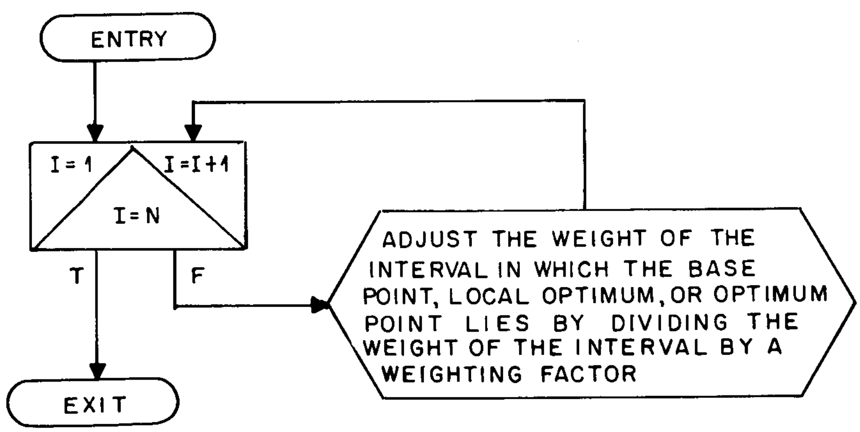
Figure F-13. Optimization Logic Diagrams (13 of 21)



ARGUMENTS	
INPUT	OUTPUT
NW	JI
NMAX	
XMAX	
DDX	
XIPT	
DX	
IFRT	
INDIC	
IC	
Z	
IEXT	
NCON	

Interval Location (LOCINT) Subroutine

Figure F-13. Optimization Logic Diagrams (14 of 21)

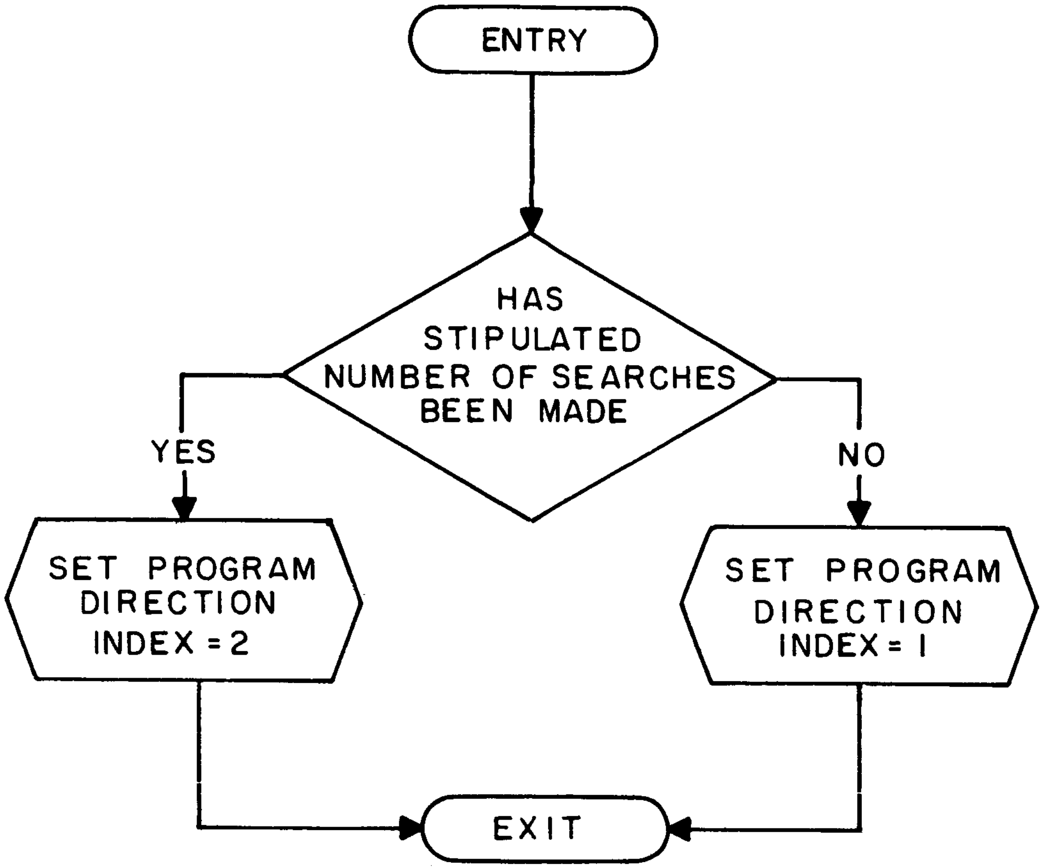


N=NO. OF INDEPENDENT VARIABLES

ARGUMENTS	
INPUT	OUTPUT
NMAX	WGT
WFAC	
JI	

Weight Adjustment (ADJUST) Subroutine

Figure F-13. Optimization Logic Diagrams (15 of 21)

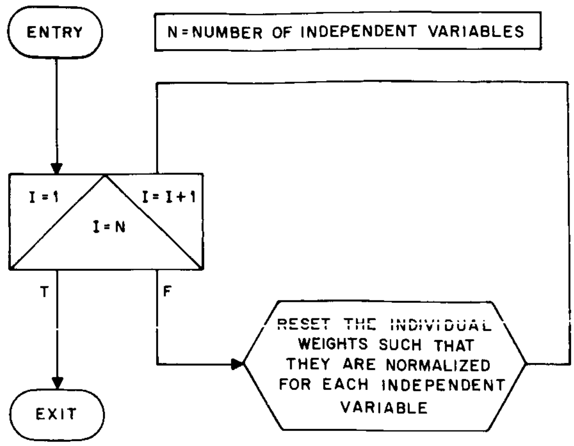


ARGUMENTS	
INPUT	OUTPUT
LS LSMAX	INDIC

PROGRAM DIRECTION INDEX	
1	NUMBER OF SEARCHES NOT EXCEEDED
2	NUMBER OF SEARCHES EXCEEDED
3	
4	
5	

Termination Test (END) Subroutine

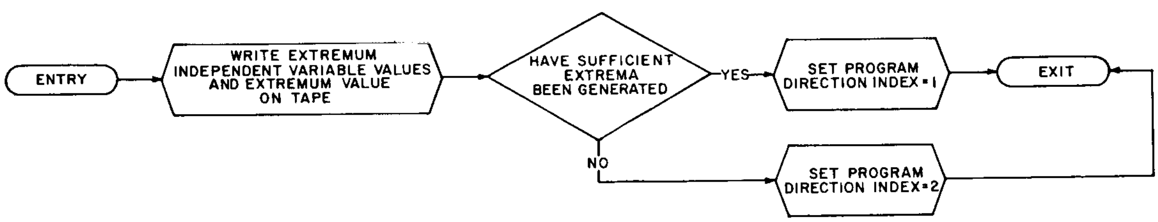
Figure F-13. Optimization Logic Diagrams (16 of 21)



ARGUMENTS	
INPUT	OUTPUT
NMAX NW DDX	WGT

Weight Normalization (NORMAL) Subroutine

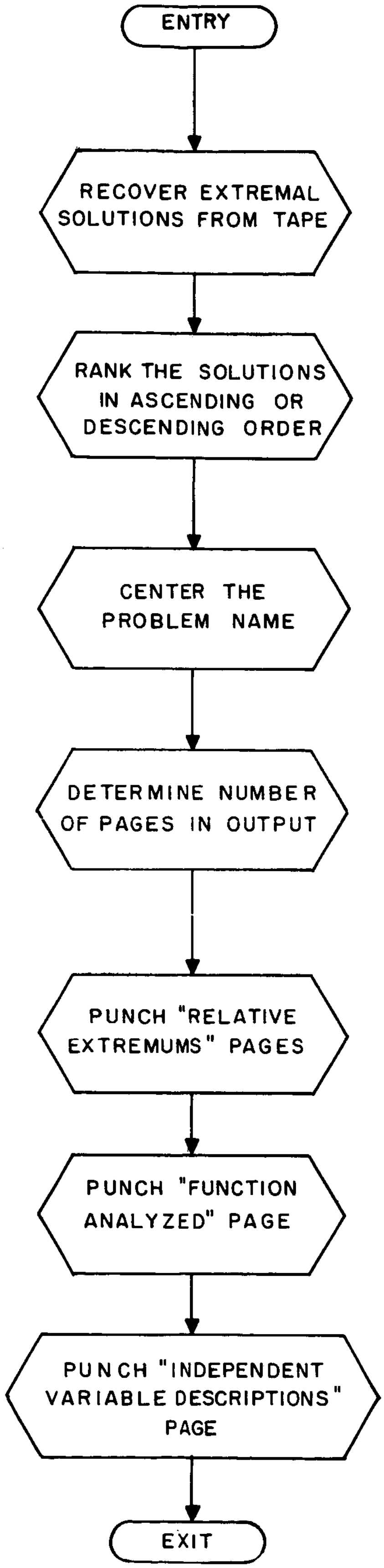
Figure F-13. Optimiation Logic Diagrams (17 of 21)



ARGUMENTS	
INPUT	OUTPUT
XIPT IEXT IC Z IFRT NCON NMAX LP LPMAX INDIC MARY	

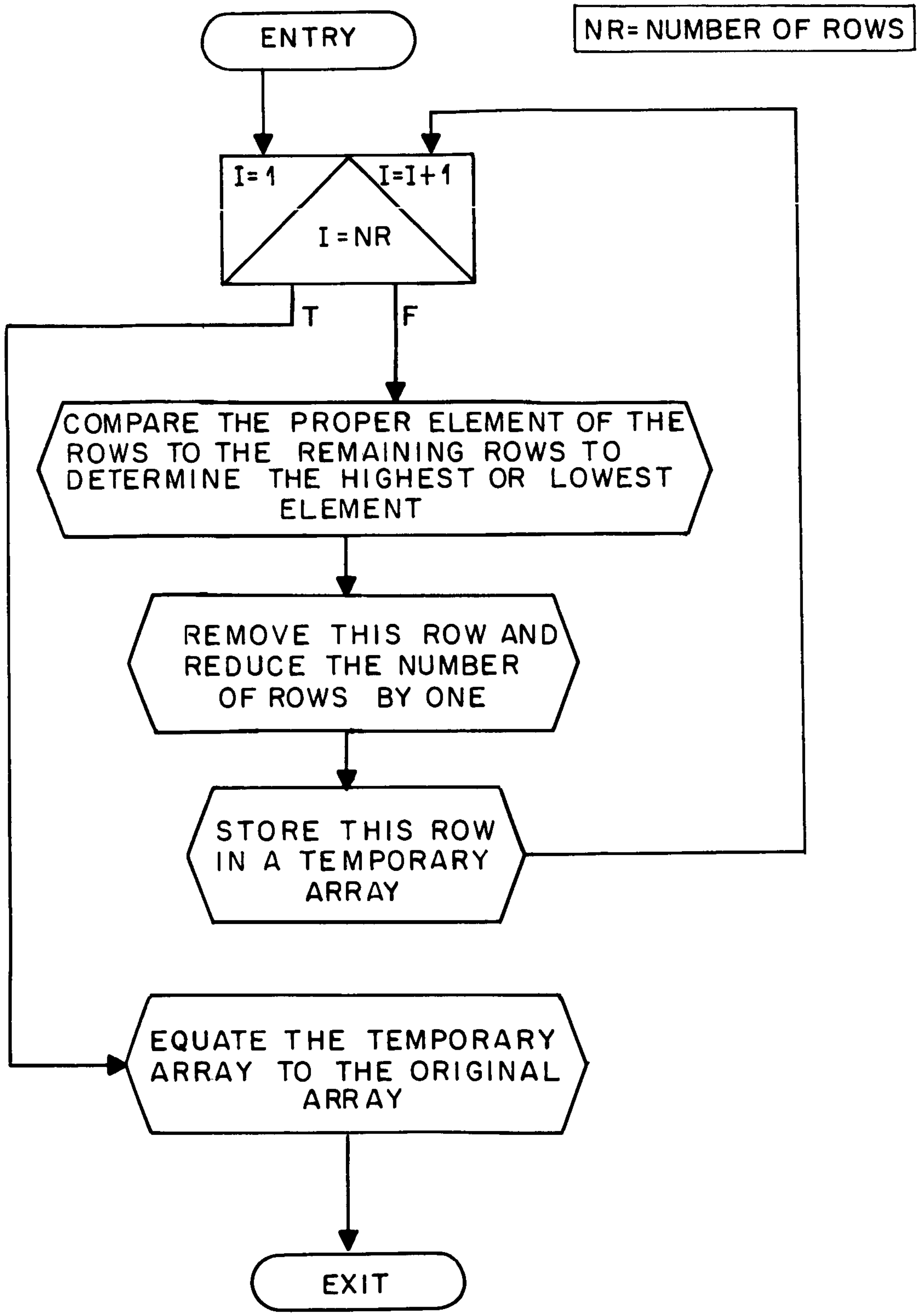
Optimal Solution Storage (STORAG) Subroutine

Figure F-13. Optimization Logic Diagrams (18 of 21)



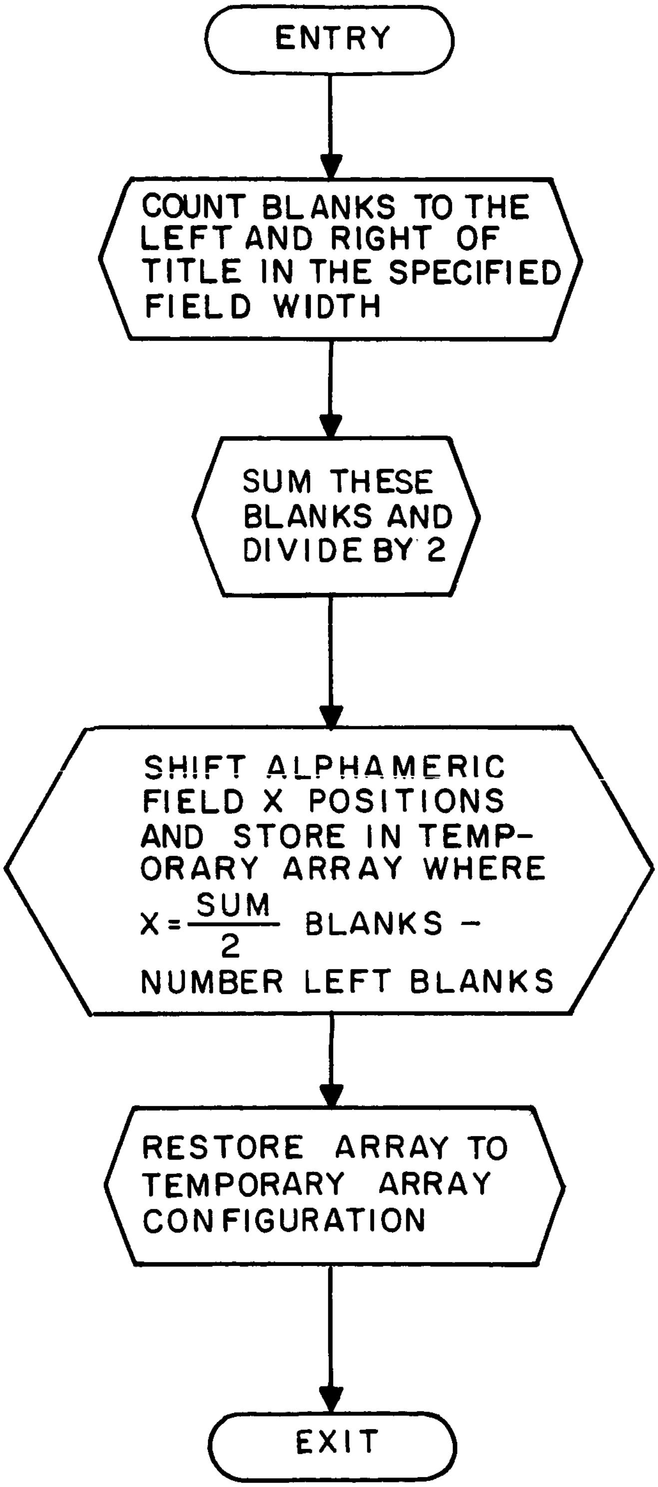
Output (OUTPUT) Subroutine

Figure F-13. Optimization Logic Diagrams (19 of 21)



Information Centering (CENTRE) Subroutine

Figure F-13. Optimization Logic Diagrams (20 of 21)



Ordering (RANK) Subroutine

Figure F-13. Optimization Logic Diagrams (21 of 21)

```
C
C                          *** PROGRAM DIRECTION INDICATOR ***
C
C     INDIC(1)=0  NO PRINT
C     INDIC(1)=1  PRINT
C     INDIC(2)=0  RANDOM POINT GENERATION
C     INDIC(2)=1  READ STARTING POINT
C     INDIC(3)=1  UNABLE TO GENERATE FEASIBLE STARTING POINT
C     INDIC(3)=2  FEASIBLE
C     INDIC(3)=3  NOT FEASIBLE
C     INDIC(4)=1  VALID SEARCH VECTOR
C     INDIC(4)=2  OPTIMUM POINT
C     INDIC(4)=3  CONSTRAINT VIOLATION
C     INDIC(5)=1  VALID EXPANDING SEARCH
C     INDIC(5)=2
C     INDIC(5)=3  BOUNDARY OR CONSTRAINT VIOLATION
C
C                                        *** PROGRAM ***
      DIMENSION YN(50),YP(50),CO(5),CN(50,5),C(50,5),ICN(50,5),ICP(50,5)
      DIMENSION XMIN(50),XMAX(50),DX(50),DY(57),DDX(50),INDIC(5)
      DIMENSION NAME(14),NPROB(40),NDATE(3),ITYPE(7),NSYM(50,3)
      DIMENSION XIPT(50),TX(50),ACC(50),NW(50),JI(50),WGT(50,20)
      DIMENSION IFUNC(10,36)
      DIMENSION IC(5),Z(5)
      EQUIVALENCE(LS,IC(1))
      COMMON NAME,NPROB,NDATE,NCARDS,IFUNC,XMIN,XMAX,NSYM,ACC,ITYPE
C     *** DATA INPUT ***
    1 CALL DATA(NMAX,GAMMA,LPMAX,LSMAX,WFAC,NCON,INDIC,NW,XIPT)
C     *** INITIALIZATION ***
    2 CALL START(IFRT,LS,LP,MARY,LAMB,NMAX,ACC,XMAX,XMIN,DX,NW,DDX,WGT)
C     *** TYPE EXTREMUM ***
    3 CALL EXTREM(ITYPE,IEXT)
C     *** INITIAL POINT SELECTION IN N-SPACE ***
    4 CALL NPOINT(INDIC,NMAX,NW,WGT,XIPT,DDX,XMIN,JI)
      KONCAV=0
    5 CALL FEASBL(MARY,LAMB,IEXT,XIPT,IC,Z,IFRT,NCON,NMAX,WGT,WFAC,JI,IN
     1DIC,NW,DDX)
C     *** FEASIBILITY TEST ***
      IND=INDIC(3)
      GO TO(99,6,4),IND
C     *** SEARCH DIRECTION DETERMINATION ***
    6 LS=LS+1
      CALL STEEP(NMAX,XIPT,DX,IEXT,DY,GAMMA,XMAX,XMIN,NCON,INDIC,YO,Y
     1N,YP,CO,CN,C,ICP,ICN,IFRT,IC,Z,TX)
      IND=INDIC(4)
      GO TO(13,10,7),IND
    7 CALL STEEP1(YO,YN,YP,ICP,ICN,XIPT,NMAX,NCON,DY,INDIC,DX,GAMMA,IEXT
     1,IC,Z,IFRT,TX)
      IND=INDIC(5)
      GO TO(8,10,13),IND
    8 CALL TANHYP(NCON,NMAX,YP,YO,ICP,ICN,INDIC,C,CO,IEXT,DY,GAMMA,XIPT,
     1DX,IC,Z,IFRT)
      IND=INDIC(4)
      GO TO(10,13,9),IND
    9 CALL TANP1(KONCAV,YO,NMAX,XIPT,DY,NCON,Z,IEXT,GAMMA,DX,XMAX,XMIN,I
```

Figure F-14. Program Listings (1 of 31)

```
     1INDIC,YN,YP,CO,CN,C,ICP,ICN,IFRT,IC,TX)
      IND=INDIC(4)
      GO TO(10,13,13),IND
C     *** SEARCH ***
   10 CALL EXPAND(GAMMA,NMAX,XMAX,XMIN,XIPT,IEXT,DY,DX,INDIC,TX,BLD
     1AN,BLDAO,BLDAP,YYN,YYO,YYP,NCON,IFRT,IC,Z)
      IND=INDIC(4)
      GO TO(11,99,12),IND
   11 CALL NARROW(BLDAO,BLDAN,BLDAP,YYO,YYN,YYP,NMAX,IEXT,TX,XIPT,DY,DX,
     1INDIC,NCON,IC,Z,IFRT)
C     *** INTERVAL LOCATION ***
   12 CALL LOCINT(NW,NMAX,XMIN,XMAX,DDX,XIPT,TX,JI,IFRT,INDIC,IC,Z,IEXT,
     1NCON)
C     *** WEIGHT ADJUSTMENT ***
   14 CALL ADJUST(NMAX,WGT,WFAC,JI)
   16 CALL END(LS,LSMAX,INDIC)
      IND=INDIC(5)
      GO TO(6,13),IND
C     *** WEIGHT NORMALIZATION ***
   13 CALL NORMAL(NMAX,NW,WGT,DDX)
C     *** EXTREMUM STORAGE ***
   15 CALL STORAG(XIPT,IEXT,IC,Z,IFRT,NCON,NMAX,LP,LPMAX,INDIC,MARY)
      IND=INDIC(5)
      GO TO(4,17),IND
C     *** OUTPUT ***
   17 CALL OUTPUT(NMAX,LP,IEXT)
      CALL SYSTEM
   99 WRITE OUTPUT TAPE 6,200
  200 FORMAT (1H ,13HPROGRAM ERROR)
      CALL SYSTEM
      END
```

Figure F-14. Program Listings (2 of 31)

```
      SUBROUTINE DATA(NMAX,GAMMA,LPMAX,LSMAX,WFAC,NCON,INDIC,NW,XIPT)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     ACC     PERCENT ACCURACY REQUIRED ON EACH INDEPENDENT VARIABLE
C     GAMMA   EXPANSION RATE FACTOR
C     IFUNC   ALPHANUMERIC DESCRIPTION OF THE FUNCTION
C     ITYPE   TYPE EXTREMUM ( MINIMUM OR MAXIMUM )
C     LPMAX   NUMBER OF RELATIVE EXTREMA DESIRED
C     LSMAX   NUMBER OF SEARCHES DESIRED
C     NAME    NAME OF ORIGINATOR
C     NCARDS  NUMBER OF FUNCTION DESCRIPTION CARDS
C     NCON    NUMBER OF CONSTRAINTS
C     NDATE   DATE  MO./DAY/YEAR
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     NN      CARD SEQUENCE NUMBER
C     NPROB   NAME ASSIGNED TO THE PROBLEM
C     NSYM    NAME ASSIGNED TO EACH INDEPENDENT VARIABLE
C     NW      NUMBER OF SUBINTERVALS ( USED IN WEIGHTING PROCEDURE )
C     XMAX    MAXIMUM VALUE OF THE INDEPENDENT VARIABLE
C     XMIN    MINIMUM VALUE OF THE INDEPENDENT VARIABLE
C     WFAC    WEIGHTING FACTOR
C
C                                   *** PROGRAM ***
C
      DIMENSION IFUNC(10,36)
      DIMENSION XMIN(50),XMAX(50),NW(50),INDIC(5),ACC(50)
      DIMENSION NAME(14),NPROB(40),NDATE(3),ITYPE(7),NSYM(50,3)
      COMMON NAME,NPROB,NDATE,NCARDS,IFUNC,XMIN,XMAX,NSYM,ACC,ITYPE,IBLK
  100 FORMAT(1X,14A1,5X,40A1,10X,2(I2,1X),I2,I2)
  101 FORMAT(1A,4X,7A1,5X,F5.2,4(5X,I2),5X,F6.2,2(5X,I2),4X,I2)
  102 FORMAT(I3,75X,I2)
  103 FORMAT(36A2,6X,I2)
  104 FORMAT(5X,3A2,5X,F10.4,5X,F10.4,5X,F10.4,5X,F6.2,5X,I4,2X,I2)
  110 FORMAT(21HERROR IN INPUT FORMAT)
    1 READ 100,IBLK,(NAME(I),I=1,14),(NPROB(I),I=1,40),(NDATE(I),I=1,3),
     1NN
      IF(NN-1)199,2,199
    2 READ 101,(ITYPE(I),I=1,7),GAMMA,NMAX,NCON,LPMAX,LSMAX,WFAC,INDIC(1
     1),INDIC(2),NN
      IF(NN-2)199,3,199
    3 READ 102,NCARDS,NN
      IF(NN-3)199,4,199
    4 DO 5 NC=1,NCARDS
      READ 103,(IFUNC(NC,I),I=1,36),NN
      IF(NN-3)199,5,199
    5 CONTINUE
      DO 9 I=1,NMAX
      READ 104,(NSYM(I,J),J=1,3),XMIN(I),XMAX(I),XIPT(I),NW(I),ACC(I),NN
      IF(NN-4)199,9,199
    9 CONTINUE
      RETURN
  199 WRITE OUTPUT TAPE 6,110
      CALL SYSTEM
      END
```

Figure F-14. Program Listings (3 of 31)

```
      SUBROUTINE START(IFRT,LS,LP,MARY,LAMB,NMAX,ACC,XMAX,XMIN,DX,NW,DDX
     1,WGT)
C
C                           *** DEFINITION OF VARIABLES ***
C
C     DDX      SUBINTERVAL SIZE
C     DX       INCREMENT OF THE INDEPENDENT VARIABLE
C     FM       SUBSTITUTION
C     FNW      SUBSTITUTION
C     IFRT     FEASIBILITY STATUS INDICATOR
C     LAMB     TERMINATION INDEX
C     LP       CURRENT EXTREMUM COUNT
C     LS       CURRENT SEARCH COUNT
C     MARY     STARTING POINT COUNT
C     NMAX     NUMBER OF INDEPENDENT VARIABLES
C     NW       NUMBER OF SUBINTERVALS OF VARIABLE RANGE
C     NWI      SUBSTITUTION
C     WGT      WEIGHT OF SUBINTERVAL OF INDEPENDENT VARIABLE
C     XD       RANGE OF INDEPENDENT VARIABLES
C     XMAX     MAXIMUM VALUE OF INDEPENDENT VARIABLE
C     XMIN     MINIMUM VALUE OF INDEPENDENT VARIABLE
C
C                                     *** PROGRAM ***
C
      DIMENSION ACC(50),XMAX(50),XMIN(50),DX(50),NW(50),DDX(50),WGT(50,2
     10)
      IFRT=0
      LS=0
      LP=0
      MARY=0
      LAMB=1000
      DO 1 I=1,NMAX
      FM=100./ACC(I)
      XD=XMAX(I)-XMIN(I)
      DX(I)=XD/FM
      FNW=NW(I)
      DDX(I)=XD/FNW
      NWI=NW(I)
      DO 2 J=1,NWI
    2 WGT(I,J)=1./XD
    1 CONTINUE
      RETURN
      END
```

Figure F-14. Program Listings (4 of 31)

```
      SUBROUTINE NPOINT(INDIC,NMAX,NW,WGT,XIPT,DDX,XMIN,JI)
C
C                        *** DEFINITION OF VARIABLES ***
C
C     DDX     SUBINTERVAL SIZE
C     DS      AREA OF SUBINTERVAL
C     FJ      SUBSTITUTION
C     INDIC   PROGRAM DIRECTION INDICATOR
C     JI      SUBINTERVAL NUMBER OF INDEPENDENT VARIABLE
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     NW      NUMBER OF SUBINTERVALS OF VARIABLE RANGE
C     NWI     SUBSTITUTION
C     RN      RANDOM NUMBER
C     S       SUMMING VARIABLE
C     WGT     WEIGHT OF SUBINTERVAL OF INDEPENDENT VARIABLE
C     XIPT    CURRENT INDEPENDENT VARIABLE VALUE
C     XMIN    MINIMUM VALUE OF INDEPENDENT VARIABLE
C
C                                 *** PROGRAM ***
C
      DIMENSION XIPT(50),XMIN(50),DDX(50)
      DIMENSION NW(50),WGT(50,20),JI(50),INDIC(5)
  101 FORMAT(1H ,14H INITIAL POINT/(3H X(,I2,3H )=,E10.4))
      IF(INDIC(2)-1)1,4,1
    1 DO 3 I=1,NMAX
      RN=RANDOM(RN)
      NWI=NW(I)+1
      S=0.
      DO 5 J=2,NWI
      DS=WGT(I,J-1)*DDX(I)
    6 K=J
      IF((S+DS)-RN)5,9,9
    5 S=S+DS
    9 K=K-1
      JI(I)=K
      FJ=K-1
      XIPT(I)=XMIN(I)+FJ*DDX(I)+(RN-S)/WGT(I,K)
    3 CONTINUE
      GO TO 4
    4 IF(INDIC(1)-1)7,8,7
    8 WRITE OUTPUT TAPE 6,101,(I,XIPT(I),I=1,NMAX)
    7 RETURN
      END
```

Figure F-14. Program Listings (5 of 31)

```
      SUBROUTINE FEASBL(MARY,LAMB,IEXT,XIPT,IC,Z,IFRT,NCON,NMAX,WGT,WFAC
     1,JI,INDIC,NW,DDX)
C
C                        *** DEFINITION OF VARIABLES ***
C
C     IFRT   FEASIBILITY STATUS INDICATOR
C     INDIC  PROGRAM DIRECTION INDICATOR
C     LAMB   TERMINATION INDEX
C     MARY   STARTING POINT COUNT
C
C                                 *** PROGRAM ***
C
      DIMENSION XIPT(50),IC(5),Z(5),WGT(50,20),JI(50),INDIC(5)
      DIMENSION NW(50),DDX(50)
      CALL FUNCY(IEXT,XIPT,Y,IC,Z,IFRT,NCON)
      CALL ADJUST(NMAX,WGT,WFAC,JI)
      MARY=MARY+1
      IF(MARY-LAMB)1,1,2
    2 INDIC(3)=1
      RETURN
    1 IF(IFRT-3)4,3,33
    3 CALL NORMAL(NMAX,NW,WGT,DDX)
      INDIC(3)=3
      RETURN
    4 INDIC(3)=2
      RETURN
   33 WRITE OUTPUT TAPE 6,100
  100 FORMAT(1H ,12HERROR FEASBL)
      END
```

Figure F-14. Program Listings (6 of 31)

```
      SUBROUTINE EXTREM(ITYPE,IEXT)
C
C                          *** DEFINITION OF VARIABLES ***
C
C     ATYPE   EXTREMUM TYPE INDICATOR
C     IEXT    TYPE EXTREMUM ( MINIMUM OR MAXIMUM )
C
C                                       *** PROGRAM ***
C
      DIMENSION ITYPE(7)
      IF(ITYPE(2)-ITYPE(4))2,1,2
    1 IEXT=0
      RETURN
    2 IEXT=1
      RETURN
      END
```

Figure F-14. Program Listings (7 of 31)

```
      SUBROUTINE STEEP(NMAX,X,DX,IEXT,DY,GAMMA,XMAX,XMIN,NCON,INDIC,YO,Y
     1N,YP,CO,CN,C,ICP,ICN,IFRT,IC,Z,TX)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     C       CONSTRAINING FUNCTION VALUE AT INCREMENT
C     CN      CONSTRAINT VALUE AT NEGATIVE INCREMENT
C     DELY    CHANGE IN OBJECTIVE FUNCTION IN DECREASING DIRECTION
C     DELYN   CHANGE IN OBJECTIVE FUNCTION AT NEGATIVE INCREMENT
C     DELYP   CHANGE IN OBJECTIVE FUNCTION AT POSITIVE INCREMENT
C     DX      INCREMENT OF THE INDEPENDENT VARIABLE
C     DY      DIRECTION NUMBER INDICATING DIRECTION OF STEEPEST DESCENT
C     GAMMA   EXPANSION RATE FACTOR
C     IC      CONSTRAINT STATUS INDICATOR
C     ICN     CONSTAINT STATUS AT NEGATIVE INCREMENT
C     ICP     CONSTAINT STATUS AT POSITIVE INCREMENT
C     IFRT    FEASIBILITY STATUS INDICATOR
C     INDIC   PROGRAM DIRECTION INDICATOR
C     NCON    NUMBER OF CONSTRAINTS
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     NULL    OPTIMALITY VARIABLE
C     R       SUBSTITUTION
C     RMSDY   SUBSTITUTION
C     SUMDY   SUM OF CHANGES IN OBJECTIVE FUNCTION
C     TX      SUBSTITUTION
C     X       CURRENT INDEPENDENT VARIABLE VALUE
C     XMAX    MAXIMUM VALUE OF INDEPENDENT VARIABLE
C     XMIN    MINIMUM VALUE OF INDEPENDENT VARIABLE
C     Y       VALUE OF THE OBJECTIVE FUNCTION AT BASE POINT
C     YN      OBJECTIVE FUNCTION VALUE AT NEGATIVE INCREMENT
C     YP      VALUE OF THE OBJECTIVE FUNCTION AT POSITIVE INCREMENT
C     Z       CURRENT CONSTRAINING FUNCTION VALUE
C
C                                   *** PROGRAM ***
C
      DIMENSION TX(50),X(50),DX(50),XMAX(50),XMIN(50),YP(50),C(50, 5),CO
     1( 5),ICP(50, 5)
      DIMENSION IC( 5),YN(50),CN(50, 5),ICN(50, 5),DELY(50),INDIC(5),DY(
     157),Z( 5)
      CALL FUNCY(IEXT,X,YO,IC,CO,IFRT,NCON)
      DO 1 I=1,NMAX
    1 TX(I)=X(I)
      DO 2 I=1,NMAX
      TX(I)=TX(I)+DX(I)
      IF(TX(I)-XMAX(I))3,3,4
    4 TX(I)=XMAX(I)
    3 CALL FUNCY(IEXT,TX,Y,IC,Z,IFRT,NCON)
      YP(I)=Y
      IF(NCON)28,27,28
   28 DO 5 J=1,NCON
      C(I,J)=Z(J)
    5 ICP(I,J)=IC(J)
   27 TX(I)=X(I)-DX(I)
      IF(TX(I)-XMIN(I))6,7,7
    6 TX(I)=XMIN(I)
```

Figure F-14. Program Listings (8 of 31)

```
  7 CALL FUNCY(IEXT,TX,Y,IC,Z,IFRT,NCON)
    YN(I)=Y
    IF(NCON)29,2,29
 29 DO 8 J=1,NCON
    CN(I,J)=Z(J)
  8 ICN(I,J)=IC(J)
  2 TX(I)=X(I)
  9 NULL=0
    DO 10 I=1,NMAX
    DELYP=YO-YP(I)
    DELYN=YO-YN(I)
    IF(DELYN)11,11,12
 11 IF(DELYP)13,13,12
 12 IF(DELYP-DELYN)14,15,15
 14 DELY(I)=-DELYN
    GO TO 10
 15 DELY(I)=DELYP
    GO TO 10
 13 DELY(I)=0.
    NULL=NULL+1
 10 CONTINUE
    IF(NMAX-NULL)17,16,17
 16 INDIC(4)=1
    RETURN
 17 SUMDY=0.
    DO 18 I=1,NMAX
 18 SUMDY=SUMDY+DELY(I)*DELY(I)
    RMSDY=SQRT(SUMDY)
    DO 19 I=1,NMAX
 19 DY(I)=DELY(I)/RMSDY
    IF(IFRT-40)26,26,25
 26 IF(INDIC(1)-1)20,21,20
 21 WRITE OUTPUT TAPE 6,101,(I,DY(I),I=1,NMAX)
101 FORMAT(1H ,3HDY(,I2,2H)=,E10.4)
 20 R=GAMMA-1.
    DO 22 I=1,NMAX
 22 TX(I)=X(I)+R*DY(I)*DX(I)
    CALL FUNCY(IEXT,TX,Y,IC,Z,IFRT,NCON)
    IF(IFRT-3)23,24,23
 23 INDIC(4)=2
    RETURN
 24 INDIC(4)=3
 25 RETURN
    END
```

Figure F-14. Program Listings (9 of 31)

```
      SUBROUTINE STEEP1(YO,YN,YP,ICP,ICN,X,NMAX,NCON,DY,INDIC,DX,GAMMA,I
     1EXT,IC,Z,IFRT,TX)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     DELY    CHANGE IN OBJECTIVE FUNCTION IN DECREASING DIRECTION
C     DELYN   CHANGE IN OBJECTIVE FUNCTION AT NEGATIVE INCREMENT
C     DELYP   CHANGE IN OBJECTIVE FUNCTION AT POSITIVE INCREMENT
C     DY      DIRECTION NUMBER INDICATING DIRECTION OF STEEPEST DESCENT
C     GAMMA   EXPANSION RATE FACTOR
C     ICN     CONSTAINT STATUS AT NEGATIVE INCREMENT
C     ICP     CONSTAINT STATUS AT POSITIVE INCREMENT
C     IFRT    FEASIBILITY STATUS INDICATOR
C     INDIC   PROGRAM DIRECTION INDICATOR
C     NG      CONSTRAINT STATUS INDICATOR
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     NULL    OPTIMALITY VARIABLE
C     R       SUBSTITUTION
C     RMSDY   SUBSTITUTION
C     SUMDY   SUM OF CHANGES IN OBJECTIVE FUNCTION
C     TX      SUBSTITUTION
C     X       CURRENT INDEPENDENT VARIABLE VALUE
C     YN      OBJECTIVE FUNCTION VALUE AT NEGATIVE INCREMENT
C     YO      OBJECTIVE FUNCTION VALUE AT BASE POINT
C     YP      VALUE OF THE OBJECTIVE FUNCTION AT POSITIVE INCREMENT
C
C                                   *** PROGRAM ***
C
      DIMENSION YP(50),YN(50),ICP(50, 5),ICN(50, 5),DELY(50),INDIC(5),TX
     1(50),X(50),DX(50),DY(57),IC( 5),Z( 5)
      NG=0
      NULL=0
      DO 1 I=1,NMAX
      DELYP=YO-YP(I)
      DELYN=YO-YN(I)
      DO 2 J=1,NCON
      IF(ICP(I,J))4,2,4
    2 CONTINUE
      GO TO 5
    4 DELYP=0.
      NG=1
    5 DO 6 J=1,NCON
      IF(ICN(I,J))7,6,7
    6 CONTINUE
      GO TO 3
    7 DELYN=0.
      NG=1
    3 IF(DELYN)8,8,9
    8 IF(DELYP)10,10,9
    9 IF(DELYP-DELYN)11,12,12
   11 DELY(I)=-DELYN
      GO TO 1
   12 DELY(I)=DELYP
      GO TO 1
   10 DELY(I)=0.
```

Figure F-14. Program Listings (10 of 31)

```
      NULL=NULL+1
    1 CONTINUE
      IF(NMAX-NULL)13,14,13
   14 IF(NG-1)15,16,15
   16 INDIC(5)=1
      RETURN
   15 INDIC(5)=3
      RETURN
   13 SUMDY=0.
      DO 17 I=1,NMAX
   17 SUMDY=SUMDY+DELY(I)*DELY(I)
      RMSDY=SQRT(SUMDY)
      DO 18 I=1,NMAX
   18 DY(I)=DELY(I)/RMSDY
      IF(INDIC(1)-1)20,19,20
   19 WRITE OUTPUT TAPE 6,101,(I,DY(I),I=1,NMAX)
  101 FORMAT(1H ,4HDY1(,I2,2H)=,E10.4)
   20 R=GAMMA-1.
      DO 21 I=1,NMAX
   21 TX(I)=X(I)+R*DY(I)*DX(I)
      CALL FUNCY(IEXT,TX,Y,IC,Z,IFRT,NCON)
      IF(IFRT-3)22,23,22
   22 INDIC(5)=2
      RETURN
   23 INDIC(5)=1
      RETURN
      END
```

Figure F-14. Program Listings (11 of 31)

```
      SUBROUTINE TANHYP(NCON,NMAX,YP,YO,ICP,ICN,INDIC,C,CO,IEXT,DY,GAMMA
     1,X,DX,IC,Z,IFRT)
C
C                          *** DEFINITION OF VARIABLES ***
C
C     CO       CONSTRAINT VALUE AT BASE POINT
C     C        CONSTRAINT VALUE
C     DX       INCREMENT OF THE INDEPENDENT VARIABLE
C     DY       DIRECTION NUMBER INDICATING DIRECTION OF STEEPEST DESCENT
C     G        LINEAR PROGRAM MATRIX ELEMENT
C     GAMMA    EXPANSION RATE FACTOR
C     IC       CONSTRAINT STATUS INDICATOR
C     ICN      CONSTAINT STATUS AT NEGATIVE INCREMENT
C     ICP      CONSTAINT STATUS AT POSITIVE INCREMENT
C     IFRT     FEASIBILITY STATUS INDICATOR
C     INDIC    PROGRAM DIRECTION INDICATOR
C     KI       SUBSTITUTION
C     KKM      SUBSTITUTION
C     KM       SUBSTITUTION
C     KMI      SUBSTITUTION
C     KR       NULL SEARCH VECTOR INDICATOR
C     LSUB     LINEAR PROGRAM VALIDITY INDICATOR
C     NCOL     NUMBER OF COLUMNS IN MATRIX (LP)
C     NCON     NUMBER OF CONSTRAINTS
C     NCON1    SUBSTITUTION
C     NCONT    SUBSTITUTION
C     NMAX     NUMBER OF INDEPENDENT VARIABLES
C     NROW     NUMBER OF ROWS IN MATRIX (LP)
C     R        SUBSTITUTION
C     TX       SUBSTITUTION
C     X        CURRENT INDEPENDENT VARIABLE VALUE
C     Y        CURRENT VALUE OF THE OBJECTIVE FUNCTION
C     YO       OBJECTIVE FUNCTION VALUE AT BASE POINT
C     YP       VALUE OF THE OBJECTIVE FUNCTION AT POSITIVE INCREMENT
C
C                                   *** PROGRAM ***
C
      DIMENSION G(57,107),IC(5),YP(50),ICP(50,5),ICN(50,5),INDIC(5)
      DIMENSION C(50,5),CO(5),DY(57),DX(50),TX(50),X(50),Z(5)
      NCON1=NCON+1
      NCONT=NCON+NMAX
      NROW=NCONT+2
      NCOL=NROW+NMAX
      DO 1 N=1,NROW
      DO 1 J=1,NCOL
    1 G(N,J)=0.
      DO 2 J=1,NCON
    2 IC(J)=0
      DO 3 J=1,NMAX
    3 G(1,J)=(YP(J)-YO)/DX(J)
      DO 4 I=1,NMAX
      DO 4 J=1,NCON
      IF(ICP(I,J))5,6,5
    6 IF(ICN(I,J))5,4,5
    5 IC(J)=1
```

Figure F-14. Program Listings (12 of 31)

```
  4 CONTINUE
    IF(INDIC(1)-1)8,7,8
  7 WRITE OUTPUT TAPE 6,100,(J,IC(J),J=1,NCON)
100 FORMAT(1H ,3HIC(,I2,2H)=,I2)
  8 K=2
    DO 9 J=1,NCON
    IF(IC(J)-1)9,10,33
 10 DO 11 M=1,NMAX
 11 G(K,M)=(C(M,J)-CO(J))/DX(M)
    KM=NMAX+K-1
    G(K,KM)=-1.
    K=K+1
  9 CONTINUE
    KKM=KM+NMAX+1
    JK=K-1
    DO 12 I=2,JK
    DO 12 J=1,NMAX
 12 G(I,KKM)=G(I,KKM)+G(I,J)
    DO 13 I=1,NMAX
    G(K,I)=1.
    KI=KM+I
    G(K,KI)=1.
    G(K,KKM)=2.
 13 K=K+1
    K=K-1
    DO 14 I=2,K
    IF(G(I,KKM))15,14,14
 15 DO 16 J=1,KKM
 16 G(I,J)=-G(I,J)
 14 CONTINUE
    KM1=KM+1
    LSUB=0
    CALL LPSUB(K,KKM,G,IEXT,DY,NMAX,LSUB)
    IF(LSUB)33,17,18
 17 DO 19 I=1,KM1
 19 DY(I)=DY(I)-1.
    IF(INDIC(1)-1)20,21,20
 21 WRITE OUTPUT TAPE 6,101,(I,DY(I),I=1,NMAX)
101 FORMAT(1H ,4HDY2(,I2,2H)=,E10.4)
 20 KR=0
    DO 22 I=1,NMAX
    IF(DY(I)-.001)23,23,22
 23 IF(DY(I)+.001)22,24,24
 24 KR=KR+1
 22 CONTINUE
    IF(KR-NMAX)25,18,33
 25 R=GAMMA-1.
    DO 26 I=1,NMAX
 26 TX(I)=X(I)+R*DY(I)*DX(I)
    CALL FUNCY(IEXT,TX,Y,IC,Z,IFRT,NCON)
    IF(IFRT-3)27,28,33
 27 IF(Y-YO)29,18,18
 29 INDIC(4)=1
    RETURN
 18 INDIC(4)=2
```

Figure F-14. Program Listings (13 of 31)

```
      RETURN
   33 WRITE OUTPUT TAPE 6,102
  102 FORMAT(1H ,12HERROR TANHYP)
      CALL SYSTEM
   28 INDIC(4)=3
      RETURN
      END
```

Figure F-14. Program Listings (14 of 31)

```
      SUBROUTINE TANP1(KONCAV,YO,NMAX,X,DY,NCON,Z,IEXT,GAMMA,DX,XMAX,XMI
     1N,INDIC,YN,YP,CO,CN,C,ICP,ICN,IFRT,IC,TX)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     DDY     SUBSTITUTION
C     DX      INCREMENT OF THE INDEPENDENT VARIABLE
C     DY      DIRECTION NUMBER INDICATING DIRECTION OF STEEPEST DESCENT
C     IEXT1   MAXIMUM INDICATOR
C     IFRT    FEASIBILITY STATUS INDICATOR
C     INDIC   PROGRAM DIRECTION INDICATOR
C     IP      MOST VIOLATING CONSTRAINT
C     KONCAV  CONCAVE CONSTRAINT INDICATOR
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     R       SUBSTITUTION
C     TX      SUBSTITUTION
C     X       CURRENT INDEPENDENT VARIABLE VALUE
C     XX      SUBSTITUTION
C     YO      OBJECTIVE FUNCTION VALUE AT BASE POINT
C     YYO     OBJECTIVE FUNCTION VALUE AT PREVIOUS CONCAVE CONDITION
C     Z       CURRENT CONSTRAINING FUNCTION VALUE
C
C                                  *** PROGRAM ***
C
      DIMENSION XX(50),X(50),DDY(50),DY(57),Z(5),DX(50),XMAX(50),XMIN(50
     1),INDIC(5),YN(50),YP(50),CO(5),CN(50,5),C(50,5),ICN(50,5),ICP(50,5
     2),IC(5),TX(50)
      IP=0
      IF(KONCAV)1,1,2
    1 YYO=YO
      KONCAV=KONCAV+1
      DO 3 I=1,NMAX
      XX(I)=X(I)
    3 DDY(I)=DY(I)
      DO 4 I=1,NCON
      IF(Z(I))5,4,4
    5 IF(IP)33,6,7
    6 IP=I
      GO TO 4
    7 IF(Z(IP)-Z(I))4,4,6
    4 CONTINUE
      IFRT=40+IP
      IEXT1=1
      CALL STEEP(NMAX,X,DX,IEXT1,DY,GAMMA,XMAX,XMIN,NCON,INDIC,YO,YN,YP,
     1CO,CN,C,ICP,ICN,IFRT,IC,Z,TX)
      R=1.
      IFRT=IFRT-40
      DO 8 I=1,NMAX
    8 TX(I)=X(I)+R*DY(I)*DX(I)
      CALL FUNCY(IEXT1,TX,Y,IC,Z,IFRT,NCON)
      IF(IFRT-3)9,10,33
    9 DO 11 I=1,NMAX
      X(I)=TX(I)
   11 DY(I)=DDY(I)
      INDIC(4)=1
```

Figure F-14. Program Listings (15 of 31)

```
      RETURN
   10 WRITE OUTPUT TAPE 6,100
  100 FORMAT(1H ,24HCHECK CONCAVE CONSTRAINT)
      INDIC(4)=3
      RETURN
   33 WRITE OUTPUT TAPE 6,101
  101 FORMAT(1H ,11HERROR TANP1)
      CALL SYSTEM
    2 IF(YO-YYO)1,14,14
   14 DO 15 I=1,NMAX
   15 X(I)=XX(I)
      INDIC(4)=2
      RETURN
      END
```

Figure F-14. Program Listings (16 of 31)

```
      SUBROUTINE LPSUB(II,JJ,A,IEXT,DY,NMAX,LSUB)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     A       LINEAR PROGRAM MATRIX ELEMENT
C     DY      DIRECTION NUMBER INDICATING DIRECTION OF STEEPEST DESCENT
C     II      NUMBER OF MATRIX ROWS
C     JJ      NUMBER OF MATRIX COLUMNS
C     L       BASIS VECTOR
C     LSUB    LINEAR PROGRAMMING VALIDITY INDICATOR
C     W       SOLUTION VECTOR
C     XMIN    PIVOT ROW SELECTOR
C
C                                   *** PROGRAM ***
C
      DIMENSION A(57,107),W(57),L(57),DY(57)
      III=II+1
      DO 30 I=1,III
      DY(I)=0.
      W(I)=0.
   30 L(I)=0
      DO 131 I=2,II
      NSLAK=NMAX+I-1
      IF(A(I,NSLAK))131,131,31
   31 L(I)=NSLAK
  131 CONTINUE
   33 KKK=0
   22 I=1
   23 I=I+1
      IF(I-III)24,40,40
   24 IF(L(I))23,25,23
   25 DO 27 J=1,JJ
      IF(A(I,J))26,27,26
   26 A(III,J)=A(III,J)-A(I,J)
   27 CONTINUE
      GO TO 23
   40 K=III
   44 J=0
      W(K)=0.0
      L(K)=0
   42 J=J+1
      IF(J-JJ)41,45,45
   41 IF(A(K,J))43,42,42
   43 IF(W(K)-A(K,J))42,42,47
   47 W(K)=A(K,J)
      L(K)=J
      GO TO 42
   45 IF(L(K))46,62,46
   46 KJ=L(K)
      DO 120 I=2,II
      IF(A(I,KJ))120,120,121
  120 CONTINUE
      GO TO 66
  121 I=1
      JK=0
```

Figure F-14. Program Listings (17 of 31)

```
 50 I=I+1
    IF(I-II)52,52,56
 52 IF(A(I,KJ))50,50,51
 51 X=A(I,JJ)/A(I,KJ)
    IF(JK)55,53,55
 55 IF(X-XMIN)53,50,50
 53 XMIN=X
    JK=I
    GO TO 50
 56 X=A(JK,KJ)
    L(JK)=KJ
    DO 57 I=1,III
 57 W(I)=A(I,KJ)
    IJ=JK-1
    DO 59 I=1,IJ
    DO 59 J=1,JJ
    IF(A(JK,J))58,59,58
 58 IF(W(I))580,59,580
580 A(I,J)=A(I,J)-W(I)*(A(JK,J)/X)
 59 CONTINUE
    IJ=JK+1
    DO 61 I=IJ,III
    DO 61 J=1,JJ
    IF(A(JK,J))60,61,60
 60 IF(W(I))600,61,600
600 A(I,J)=A(I,J)-W(I)*(A(JK,J)/X)
 61 CONTINUE
    DO 205 J=1,JJ
205 A(JK,J)=A(JK,J)/X
    KKK=KKK+1
    GO TO 44
 62 IF(K-1)70,70,63
 63 IJ=JJ-1
    DO 65 J=1,IJ
    IF(A(K,J)-.0001)65,65,66
 65 CONTINUE
    DO 230 J=1,JJ
230 A(III,J)=0.0
    K=1
    KKK=0
    GO TO 44
 70 DO 71 I=2,II
    K=L(I)
 71 DY(K)=A(I,JJ)
 76 RETURN
 66 LSUB=5
    GO TO 76
    END
```

Figure F-14. Program Listings (18 of 31)

```
      SUBROUTINE EXPAND(GAMMA,NMAX,XMAX,XMIN,XIPT,IEXT,DY,DX,INDIC,
     1TX,BLDAN,BLDAO,BLDAP,YN,YO,YP,NCON,IFRT,IC,Z)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     BLAMDA EXPANSION FACTOR
C     BLDAN  LOWER CONFINEMENT OF LOCAL EXTREMUM
C     BLDAO  LAMBDA VALUE AT ONE STEP BEFORE TERMINATION
C     BLDAP  UPPER CONFINEMENT OF LOCAL EXTREMUM
C     CLAMDA LAMBDA VALUE AT BOUNDARY
C     DX     INCREMENT OF THE INDEPENDENT VARIABLE
C     DY     DIRECTION NUMBER INDICATING DIRECTION OF STEEPEST DESCENT
C     GAMMA  EXPANSION RATE FACTOR
C     IFRT   FEASIBILITY STATUS INDICATOR
C     INDIC  PROGRAM DIRECTION INDICATOR
C     LIP    BOUNDARY VIOLATION INDICATOR
C     NMAX   NUMBER OF INDEPENDENT VARIABLES
C     SLAMDA MINIMUM LAMBDA VALUE AT BOUNDARY
C     TX     SUBSTITUTION
C     XIPT   CURRENT INDEPENDENT VARIABLE VALUE
C     XMAX   MAXIMUM VALUE OF INDEPENDENT VARIABLE
C     XMIN   MINIMUM VALUE OF INDEPENDENT VARIABLE
C     Y      OBJECTIVE FUNCTION VALUE AT N-TH STEP
C     YN     OBJECTIVE FUNCTION VALUE AT N-2
C     YO     OBJECTIVE FUNCTION VALUE AT N-1
C     YP     OBJECTIVE FUNCTION VALUE AT N
C
C                                   *** PROGRAM ***
C
      DIMENSION INDIC(5),XMAX(50),XMIN(50),XIPT(50),DY(57),DX(50),Y(100)
     1,TX(50),BLAMDA(100),IC( 5),Z( 5)
      IF(INDIC(1)-1)2,1,2
    1 WRITE OUTPUT TAPE 6,100
  100 FORMAT(1H ,16HEXPANDING SEARCH)
    2 DO 3 N=1,99
      LIP=0
      BLAMDA(N)=(GAMMA**(N-1))-1.
      DO 4 I=1,NMAX
      TX(I)=XIPT(I)+BLAMDA(N)*DY(I)*DX(I)
      IF(TX(I)-XMAX(I))5,5,6
    5 IF(TX(I)-XMIN(I))7,4,4
    7 CLAMDA=(XMIN(I)-XIPT(I))/(DY(I)*DX(I))
      GO TO 8
    6 CLAMDA=(XMAX(I)-XIPT(I))/(DY(I)*DX(I))
    8 LIP=LIP+1
      IF (LIP-2) 9,34,34
 34   IF (SLAMDA-CLAMDA) 4,9,9
    9 SLAMDA=CLAMDA
    4 CONTINUE
      IF(LIP-1)10,11,11
   11 DO 12 I=1,NMAX
      TX(I)=XIPT(I)+SLAMDA*DY(I)*DX(I)
   12 XIPT(I)=TX(I)
      IF(INDIC(1)-1)14,13,14
   13 WRITE OUTPUT TAPE 6,101,SLAMDA
```

Figure F-14. Program Listings (19 of 31)

```
 101 FORMAT(1H ,7HSLAMDA=,E10.4)
     WRITE OUTPUT TAPE 6,102,(I,XIPT(I),I=1,NMAX)
 102 FORMAT(1H ,2HX(,I2,2H)=,E10.4)
  14 CALL FUNCY(IEXT,TX,YN,IC,Z,IFRT,NCON)
     IF(IFRT-3)15,16,33
  16 IF(N-3)17,18,18
  10 CALL FUNCY(IEXT,TX,YN,IC,Z,IFRT,NCON)
  15 Y(N)=YN
     IF(IFRT-3)19,20,33
20   IF(N-3) 25,21,21
  19 IF(INDIC(1)-1)22,23,22
  23 WRITE OUTPUT TAPE 6,103,N,Y(N)
 103 FORMAT(1H ,2HY(,I2,2H)=,E10.4)
  22 IF(LIP-1)24,25,25
  24 IF(N-3)3,26,26
  26 IF(Y(N)-Y(N-1))27,21,21
  27 IF(LIP-1)3,25,25
   3 CONTINUE
     INDIC(4)=2
     RETURN
  33 WRITE OUTPUT TAPE 6,104
 104 FORMAT(1H ,12HERROR EXPAND)
     CALL SYSTEM
  17 DO 28 I=1,NMAX
28   XIPT(I)=TX(I)-SLAMDA*DY(I)*DX(I)
  25 INDIC(4)=3
     RETURN
  18 INDIC(4)=3
     DO 29 I=1,NMAX
     XIPT(I)=XIPT(I)-SLAMDA*DY(I)*DX(I)
  29 XIPT(I)=XIPT(I)+BLAMDA(N-1)*DY(I)*DX(I)
     RETURN
  21 BLDAN=BLAMDA(N-2)
     BLDAO=BLAMDA(N-1)
     BLDAP=BLAMDA(N)
     YN=Y(N-2)
     YO=Y(N-1)
     YP=Y(N)
     DO 30 I=1,NMAX
  30 TX(I)=XIPT(I)+BLAMDA(N-1)*DY(I)*DX(I)
     INDIC(4)=1
     RETURN
     END
```

Figure F-14. Program Listings (20 of 31)

```
      SUBROUTINE NARROW(BLDAO,BLDAN,BLDAP,YO,YN,YP,NMAX,IEXT,TX,XIPT,DY,
     1DX,INDIC,NCON,IC,Z,IFRT)
C
C                             *** DEFINITION OF VARIABLES ***
C
C     BLDAN   LOWER CONFINEMENT OF LOCAL EXTREMUM
C     BLDAO   LAMBDA 0 VALUE
C     BLDAP   UPPER CONFINEMENT OF LOCAL EXTREMUM
C     DB      SPAN BETWEEN LAMBDA 0 AND LAMBDA +
C     DY      DIRECTION NUMBER INDICATING DIRECTION OF STEEPEST DESCENT
C     IFRT    FEASIBILITY STATUS INDICATOR
C     INDIC   PROGRAM DIRECTION INDICATOR
C     ISTOP   SUBPROGRAM DIRECTION INDICATOR
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     TBN     LAMBDA 0 + LAMBDA - /2
C     TBP     LAMBDA 0 + LAMBDA + /2
C     TDB     CONTRACTING SEARCH TERMINATION VARIABLE
C     TT      SUBSTITUTION
C     XX      INDEPENDENT VARIABLE VALUE AT HALF
C     YN      OBJECTIVE FUNCTION VALUE AT LAMBDA -
C     YO      OBJECTIVE FUMCTION VALUE AT LAMBDA 0
C     YP      OBJECTIVE FUNCTION VALUE AT LAMBDA +
C     YTTN    CURREUT OBJECTIVE FUNCTION VALUE AT NEGATIVE HALF
C     YTTP    CURREUT OBJECTIVE FUNCTION VALUE AT POSITIVE HALF
C
C                                        *** PROGRAM ***
C
      DIMENSION TX(50),IC(5),Z(5),DY(57),XX(50),XIPT(50),DX(50),INDIC(5)
      ISTOP=0
    1 TBN=(BLDA0+BLDAN)/2.
      TBP=(BLDA0+BLDAP)/2.
      DB=BLDAP-BLDAO
      DO 2 I=1,NMAX
      IF(DY(I))3,2,3
    3 TDB=ABSF(1./DY(I))
      IF(DB-TDB)2,2,4
    2 CONTINUE
      IF(ISTOP-1)5,6,7
    5 ISTOP=1
    4 DO 8 I=1,NMAX
    8 XX(I)=XIPT(I)+TBN*DY(I)*DX(I)
      CALL FUNCY(IEXT,XX,YTTN,IC,Z,IFRT,NCON)
      IF(IFRT-3)9,10,7
    9 DO 11 I=1,NMAX
   11 XX(I)=XIPT(I)+TBP*DY(I)*DX(I)
      CALL FUNCY(IEXT,XX,YTTP,IC,Z,IFRT,NCON)
      IF(IFRT-3)12,13,7
   13 IF(YTTN-YTTP)14,14,15
   12 IF(YTTN-YTTP)14,14,17
   14 IF(YTTN-YO)16,15,15
   16 TT=YO
      YO=YTTN
      YP=TT
      TB=BLDAO
      BLDAO=TBN
```

Figure F-14. Program Listings (21 of 31)

```
      BLOAP=TB
      GO TO 1
   10 DO 18 I=1,NMAX
   18 XIPT(I)=XIPT(I)+BLDAN*DY(I)*DX(I)
      INDIC(4)=1
      CALL FUNCY(IEXT,XIPT,Y,IC,Z,IFRT,NCON)
      IF(IFRT-3)19,20,7
   20 DO 21 I=1,NMAX
   21 XIPT(I)=TX(I)
      RETURN
   17 IF(YTTP-YO)22,15,15
   22 TT=YO
      YO=YTTP
      YN=TT
      TB=BLDAO
      BLDAO=TBP
      BLDAN=TB
      GO TO 1
   15 BLDAN=TBN
      BLDAP=TBP
      YN=YTTN
      YP=YTTP
      GO TO 1
    6 DO 23 I=1,NMAX
   23 XIPT(I)=XIPT(I)+BLDAO*DY(I)*DX(I)
      INDIC(4)=1
   19 IF(INDIC(1)-1)24,25,24
   25 WRITE OUTPUT TAPE 6,100,(I,XIPT(I),I=1,NMAX)
      WRITE OUTPUT TAPE 6,101,YO
   24 RETURN
    7 WRITE OUTPUT TAPE 6,102
      CALL SYSTEM
  100 FORMAT(1H ,2HX(,I2,2H)=,E10.5)
  101 FORMAT(1H ,27HY OUTPUT CONTRACTING SEARCH,E12.6)
  102 FORMAT(1H ,12HERROR NARROW)
      END
```

Figure F-14. Program Listings (22 of 31)

```
      SUBROUTINE LOCINT(NW,NMAX,XMIN,XMAX,DDX,XIPT,TX,JI,IFRT,INDIC,IC,Z
     1,IEXT,NCON)
C
C                           *** DEFINITION OF VARIABLES ***
C
C     DDX     SUBINTERVAL SIZE
C     DS      SUBSTITUTION
C     DX      INCREMENT OF THE INDEPENDENT VARIABLE
C     IFRT    FEASIBILITY STATUS INDICATOR
C     INDIC   PROGRAM DIRECTION INDICATOR
C     JI      SUBINTERVAL NUMBER OF INDEPENDENT VARIABLE
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     NW      NUMBER OF SUBINTERVALS OF VARIABLE RANGE
C     NWI     SUBSTITUTION
C     TS      SUMMING VARIABLE
C     TX      SUBSTITUTION
C     XIPT    CURRENT INDEPENDENT VARIABLE VALUE
C     XMAX    MAXIMUM VALUE OF INDEPENDENT VARIABLE
C     XMIN    MINIMUM VALUE OF INDEPENDENT VARIABLE
C
C                                      *** PROGRAM ***
C
      DIMENSION NW(50),XMIN(50),XMAX(50),XIPT(50),JI(50),IC(5),Z(5),TX
     1(50),DDX(50),INDIC(5)
      DO 1 I=1,NMAX
      NWI=NW(I)+1
      TS=XMIN(I)
      DS=0.
      DO 2 J=1,NWI
      TS=TS+DS
      DS=DDX(I)
      K=J
      IF(TS-XIPT(I))2,3,6
    2 CONTINUE
    3 IF(XIPT(I)-XMAX(I))5,6,33
    5 IF(XIPT(I)-XMIN(I))33,7,6
    7 JI(I)=K
      GO TO 1
    6 JI(I)=K-1
    1 CONTINUE
      CALL FUNCY(IEXT,XIPT,YE,IC,Z,IFRT,NCON)
      IF(IFRT-3)8,9,33
    9 DO 10 I=1,NMAX
   10 XIPT(I)=TX(I)
      IF(INDIC(1)-1)8,11,8
   11 WRITE OUTPUT TAPE 6,100,(I,XIPT(I),I=1,NMAX)
  100 FORMAT(1H ,2HX(,I2,2H)=,E10.5)
    8 RETURN
   33 WRITE OUTPUT TAPE 6,101
  101 FORMAT(1H ,12HERROR LOCINT)
      CALL SYSTEM
      END
```

Figure F-14. Program Listings (23 of 31)

```
      SUBROUTINE ADJUST(NMAX,WGT,WFAC,JI)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     JI      SUBINTERVAL NUMBER OF INDEPENDENT VARIABLE
      JIM     SUBSTITUTION
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     WFAC    WEIGHTING FACTOR
C     WGT     WEIGHT OF SUBINTERVAL OF INDEPENDENT VARIABLE
C
C                                   *** PROGRAM ***
C
      DIMENSION WGT(50,20),JI(50)
      DO 30 I=1,NMAX
      JIM=JI(I)
   30 WGT(I,JIM)=WGT(I,JIM)/WFAC
      RETURN
      END
```

Figure F-14. Program Listings (24 of 31)

```
      SUBROUTINE NORMAL(NMAX,NW,WGT,DDX)
C
C                          *** DEFINITION OF VARIABLES ***
C
C     DDX     SUBINTERVAL SIZE
C     DSWGT   SUBSTITUTION
C     NMAX    NUMBER OF INDEPENDENT VARIABLES
C     NW      NUMBER OF SUBINTERVALS OF VARIABLE RANGE
C     NWI     SUBSTITUTION
C     SWGT    SUMMING VARIABLE
C     WGT     WEIGHT OF SUBINTERVAL OF INDEPENDENT VARIABLE
C
C                                   *** PROGRAM ***
C
      DIMENSION NW(50),WGT(50,20),DDX(50)
      DO 502 I=1,NMAX
      SWGT=0.
      NWI=NW(I)
      DO 501 J=1,NWI
      DSWGT=WGT(I,J)
  501 SWGT=SWGT+DSWGT
      DO 502 J=1,NWI
      WGT(I,J)=WGT(I,J)/(DDX(I)*SWGT)
  502 CONTINUE
      RETURN
      END
```

Figure F-14. Program Listings (25 of 31)

```
      SUBROUTINE STORAG(XIPT,IEXT,IC,Z,IFRT,NCON,NMAX,LP,LPMAX,INDIC,MAR
     1Y)
C
C                         *** DEFINITION OF VARIABLES ***
C
C      IEXT    EXTREMUM TYPE INDICATOR (MAXIMUM OR MINIMUM)
C      INDIC   PROGRAM DIRECTION INDICATOR
C      LP      CURRENT EXTREMUM COUNT
C      LPMAX   MAXIMUM NUMBER OF EXTREMA
C      XIPT    OPTIMAL INDEPENDENT VARIABLE VALUE
C      Y       OPTIMAL OBJECTIVE FUNCTION VALUE
C
C                                   *** PROGRAM ***
C
      DIMENSION INDIC(5),XIPT(50),IC( 5),Z( 5)
      CALL FUNCY(IEXT,XIPT,Y,IC,Z,IFRT,NCON)
      IF(IEXT-1)9,10,9
   10 Y=-Y
  100 FORMAT(1H ,2HX(,I2,2H)=,E10.5)
  101 FORMAT(1H ,2HY=,E12.7)
    9 IF(INDIC(1)-1)8,7,8
    7 WRITE OUTPUT TAPE6,100,(I,XIPT(I),I=1,NMAX)
      WRITE OUTPUT TAPE 6,101,Y
    8 IF(LP)5,4,5
    4 REWIND 9
    5 LP=LP+1
      WRITE TAPE 9,(XIPT(I),I=1,NMAX),Y
      IF(LP-LPMAX)1,2,2
    1 INDIC(5)=1
      GO TO 3
    2 INDIC(5)=2
      END FILE 9
    2 INDIC(5)=2
    3 MARY=0
      RETURN
      END
```

Figure F-14. Program Listings (26 of 31)

```
      SUBROUTINE END(LS,LSMAX,INDIC)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     INDIC   PROGRAM DIRECTION INDICATOR
C     LS      CURRENT SEARCH COUNT
C     LSMAX   MAXIMUM NUMBER OF SEARCHES
C     MARY    CURRENT NUMBER OF STARTING POINTS
C
C                                    *** PROGRAM ***
C
      DIMENSION INDIC(5)
      IF(LS-LSMAX)1,2,2
    1 INDIC(5)=1
      GO TO 3
    2 INDIC(5)=2
    3 RETURN
      END
```

Figure F-14. Program Listings (27 of 31)

```
      SUBROUTINE OUTPUT(NMAX,LP,IEXT)
C
C                         *** DEFINITION OF VARIABLES ***
C
C     ACC      ACCURACY DESIRED IN THE INDEPENDENT VARIABLE (PERCENT)
C     IDATE    DATE MO./DAY/YEAR
C     IEXT     EXTREMUM TYPE INDICATOR (MAXIMUM OR MINIMUM)
C     IFUNC    THE PROBLEM DESCRIPTION IN ALPHAMERIC FORMAT
C     ITYPE    EXTREMUM TYPE (MAXIMUM OR MINIMUM) IN ALPHAMERIC FORMAT
C     JEFF     NUMBER OF SOLUTIONS PER PAGE
C     JEN      MINIMUM NUMBER OF LINES REQUIRED
C     LP       NUMBER OF SOLUTIONS REQUESTED
C     NAME     NAME OF ORIGINATOR
C     NAT      TEST VARIABLE
C     NMAX     NUMBER OF INDEPENDENT VARIABLES
C     NMAX1    DEPENDENT VARIABLE POSITION IN THE ARRAY
C     NPAGE    CURRENT PAGE NUMBER
C     NOPGS    TOTAL NUMBER OF PAGES
C     NPROB    NAME GIVEN TO THE PROBLEM
C     NCARDS   NUMBER OF CARDS REQUIRED TO DESCRIBE THE PROBLEM
C     NSPACE   SPACING VARIABLE
C     NSYM     SYMBOL ATTACHED TO THE INDEPENDENT VARIABLE
C     X        RANKED (RELATIVE EXTREMUMWISE) INDEPENDENT VARIABLES
C     XIPT     INDEPENDENT VARIABLE VALUE AT THE RELATIVE EXTREMUM
C     XMAX     MAXIMUM VALUE OF THE INDEPENDENT VARIABLE
C     XMIN     MINIMUM VALUE OF THE INDEPENDENT VARIABLE
C
C                                   *** PROGRAM ***
C
      DIMENSION XIPT(50),X(50,50)
      COMMON NAME,NPROB,NDATE,NCARDS,IFUNC,XMIN,XMAX,NSYM,ACC,ITYPE,IBLK
      REWIND TAPE9
      NMAX1=NMAX+1
      DO 1 J=1,LP
      READ TAPE 9,(XIPT(I),I=1,NMAX1)
      DO 2 I=1,NMAX1
    2 X(J,I)=XIPT(I)
    1 CONTINUE
      CALL RANK(LP,X,NMAX1,IEXT)
      CALL CENTRE(NPROB,40,IBLK)
  117 FORMAT(13X,I3,4X,2HY=,E10.4)
  118 FORMAT(42X,2HX(,I2,2H)=,E10.4,4X,3A2)
  119 FORMAT(1H )
  124 FORMAT(9H1GREAT-M5,14X,34H*** FUNCTION EXTREMUM ANALYSIS ***,8X,4H
     1PAGE,I3,3H OF,I3/1X,14A1,5X,40A1,5X,4HDATE,1X,I2,1H/,I2,1H/,I2/36X
     2,7A1)
  127 FORMAT(2X,76H*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*-
     1-*--*--*--*--*--*--*--*)
  128 FORMAT(28X,23H** FUNCTION ANALYZED **)
  130 FORMAT(/20X,39H** INDEPENDENT VARIABLE DESCRIPTIONS **/9X,8HVARIAB
     1LE,6X,6HSYMBOL,6X,23H---------RANGE---------,7X,7HPERCENT/11X,4HNA
     2ME,21X,7HMINIMUM,7X,7HMAXIMUM,9X,5HERROR/)
  133 FORMAT(10X,3A2,7X,2HX(,I2,1H),6X,E10.4,3X,E10.4,7X,F6.2)
  134 FORMAT(28X,24H** RELATIVE EXTREMUMS **/12X,6HNUMBER,4X,9HDEPENDENT
     1,16X,11HINDEPENDENT,4X,6HSYMBOL/23X,8HVARIABLE,17X,9HVARIABLES/)
```

Figure F-14. Program Listings (28 of 31)

```
154 FORMAT(4X,36A2)
153 FORMAT(/37X,6HNOTES.)
    JEFF=51/NMAX1
    JEN=LP/2
    DO 7 I=1,JEN
    NAT=JEFF*I
    IF(LP-NAT)9,8,7
  7 CONTINUE
  9 NOPGS=(LP/JEFF)+3
    GO TO 10
  8 NOPGS=(LP/JEFF)+2
 10 NPAGE=1
    PUNCH 124,NPAGE,NOPGS,(NAME(I),I=1,14),(NPROB(I),I=1,40),(NDATE(I)
   1,I=1,3),(ITYPE(I),I=1,7)
    PUNCH 127
    PUNCH 134
    NB=0
    DO 3 J=1,LP
    NB=NB+1
    PUNCH 117,J,X(J,NMAX1)
    DO 4 I=1,NMAX
  4 PUNCH 118,I,X(J,I),(NSYM(I,K),K=1,3)
    IF(JEFF-NB)3,6,3
  6 NPAGE=NPAGE+1
    PUNCH 124,NPAGE,NOPGS,(NAME(I),I=1,14),(NPROB(I),I=1,40),(NDATE(I)
   1,I=1,3),(ITYPE(I),I=1,7)
    PUNCH 127
    PUNCH 134
    NB=0
  3 CONTINUE
    NPAGE=NPAGE+1
    PUNCH 124,NPAGE,NOPGS,(NAME(I),I=1,14),(NPROB(I),I=1,40),(NDATE(I)
   1,I=1,3),(ITYPE(I),I=1,7)
    PUNCH 127
    PUNCH 128
    NSPACE=(10-NCARDS)/2
    NSPACF=10-NSPACE-NCARDS
    IF(NSPACE)11,12,11
 11 DO 13 I=1,NSPACE
 13 PUNCH 119
 12 DO 14 NC=1,NCARDS
 14 PUNCH 154,(IFUNC(NC,I),I=1,36)
    IF(NSPACF)15,16,15
 15 DO 17 I=1,NSPACF
 17 PUNCH 119
 16 PUNCH 127
    PUNCH 153
    NPAGE=NPAGE+1
    PUNCH 124,NPAGE,NOPGS,(NAME(I),I=1,14),(NPROB(I),I=1,40),(NDATE(I)
   1,I=1,3),(ITYPE(I),I=1,7)
    PUNCH 127
    PUNCH 130
    DO 18 I=1,NMAX
 18 PUNCH 133,(NSYM(I,J),J=1,3),I,XMIN(I),XMAX(I),ACC(I)
    RETURN
    END
```

Figure F-14. Program Listings (29 of 31)

```
      SUBROUTINE CENTRE(IFIELD,LENGTH,IBLK)
      DIMENSION IFIELD(80),JFIELD(80)
      NVL=0
      NVR=0
      DO 31 I=1,LENGTH
      IF(IFIELD(I)-IBLK)35,32,35
   32 NVL=NVL+1
   31 CONTINUE
   35 II=I
      I=LENGTH
      DO 36 K=1,LENGTH
      IF(IFIELD(I))39,37,39
   37 NVR=NVR+1
   36 I=I-1
   39 NV2=(NVL+NVR)/2
      KK=LENGTH-(NVL-NVR)
      K=KK+NV2
   46 DO 56 I=1,LENGTH
   56 JFIELD(I)=0
      NV2=NV2+1
      DO 38 I=NV2,LENGTH
      JFIELD(I)=IFIELD(II)
      IF(II-LENGTH)38,44,44
   38 II=II+1
   44 DO 45 I=1,LENGTH
   45 IFIELD(I)=0
      DO 47 I=1,LENGTH
   47 IFIELD(I)=JFIELD(I)
      RETURN
      END
```

Figure F-14. Program Listings (30 of 31)

```
      SUBROUTINE RANK(LP,X,NMAX1,IEXT)
      DIMENSION XX(50,50),X(50,50)
      LPP=LP-1
      KP=1
      XX(KP,NMAX1)=X(1,NMAX1)
      IF(LP-1)3,20,3
    3 DO 9 J=1,LPP
      IF(IEXT)5,6,5
    6 IF(X(J+1,NMAX1)-XX(KP,NMAX1))7,7,9
      IF(X(J+1,NMAX1)-XX(KP,NMAX1))9,7,7
    7 DO 8 I=1,NMAX1
    8 XX(KP,I)=X(J+1,I)
      LL=J+1
    9 CONTINUE
      IF(XX(KP,NMAX1)-X(1,NMAX1))10,11,10
   11 DO 12 I=1,NMAX1
   12 XX(KP,I)=X(1,I)
      LL=1
   10 IF(LPP-1)19,18,19
   18 IF(LL-1)20,19,20
   19 L1=LPP+1
      IF(LL-L1)21,22,21
   21 DO 23 K=LL,LPP
      DO 24 I=1,NMAX1
   24 X(K,I)=X(K+1,I)
   23 CONTINUE
   22 LPP=LPP-1
      KP=KP+1
      GO TO 3
   20 KP=LP
      DO 25 I=1,NMAX1
   25 XX(KP,I)=X(1,I)
      DO 26 J=1,LP
      DO 27 I=1,NMAX1
   27 X(J,I)=XX(J,I)
   26 CONTINUE
      RETURN
      END
```

Figure F-14. Program Listings (31 of 31)